

GTA SA Modding book



This documentation covers important points on how the GTA SA game engine works and the job performed by various internal files. Arguably the greatest feature of the GTA SA Engine is its ability to render different weathers and permit advanced scripting with a lot of possibilities. Scripters can make complex missions: 1640 opcodes are available to use in scripts. IFP files allow the creation of animations which can be applied to pedestrians with SCM opcodes either in cutscenes or map objects. The player's wanted level can reach a maximum of six stars, similar to the previous installments GTA III and GTA: Vice City. In contrast to Vice City though where the player character drowns immediately in water, swimming is possible in GTA SA. The engine can render various effects. Vehicles are destructible and cars tunable. Garages and interiors can be added in IPL files. The player's money can range from (-10^9+1) to (10^9-1) . The game features various weapons and upto 10 can be carried at one time. Some weapons are mutually exclusive; for example the shovel and the baseball bat. Advanced path files control routes, including traffic lights for vehicles and pedestrians. The game map is subdivided into various zones whose upper and lower extents are specified by coordinates in IPL files. Models used for police peds and vehicles depend on the town zones of which there are 3 (LS, LV and SF in standard game). GTA SA uses a compact save file format that stores the most important data like weather, global variables, player info and so forth. The radar contains a lot of information and helps the player find points of interest on the map which are generated through SCM scripts.

Author: **fastman92**

Book version: **1.0**

Started on: 11 June 2011

Last updated: 4 February 2015

To add before the release:

- **0003_shake_camera.avi** to 0003 opcode
 - GTA SA probably CLEO requires Microsoft C++ Run-time
 - Replaced objects into tables up to 0019 opcode
-

Table of contents

Table of contents.....	4
General.....	10
Limitations.....	10
Primary loaded files.....	17
A game sequence of working.....	22
ASCII table.....	22
Virtual key codes.....	29
GTASA_CRC32 hashing algorithm.....	36
Environment.....	38
The basic principles.....	38
Units.....	38
Distance.....	38
Basic map file - external files to load.....	38
IMG.....	39
IDE.....	39
COLLISION.....	39
IPL and MAPZONE.....	39
TEXDICTION.....	39
MODELFIL.....	40
HIERFIL.....	40
EXIT.....	40
Declaring objects (IDE and dynamic object files).....	41
IDE files - item definition.....	41
CARS.....	41
PATH.....	43
General appearance.....	44
EnbSeries.....	44
Introduction.....	44
PROXY.....	45
GLOBAL.....	46
EFFECT.....	47
See more informations for ENGINE (to do).....	48
MOTIONBLUR (to do).....	48

PERPIXELLIGHTING (to do).....	48
INPUT.....	48
ASCII table.....	49
REFLECTION.....	56
BLOOM.....	62
SSAO (to do).....	63
COLORCORRECTION (to do).....	63
WATER.....	63
SHADOW.....	65
ENGINE (to do).....	67
MOTIONBLUR (to do).....	67
PERPIXELLIGHTING (to do).....	67
DEPTHOFFIELD.....	67
Vehicles.....	70
Definition.....	70
Types.....	70
Configuring vehicle.....	72
Declaring vehicle in IDE.....	72
Vehicle handling - handling.cfg.....	72
File format.....	72
Explanation of sections and parameters.....	72
Standard Data.....	72
Sound properties - VehicleAudioSettings.cfg.....	75
Line parameters.....	76
RadioNum.....	77
Example.....	77
Binary file specifications.....	77
IMG archive (.img).....	77
Alci`s IMG Editor.....	79
Crazy IMG Editor.....	80
G-IMG.....	81
IMG Manager.....	82
IMG Tool.....	83
Spark IMG Editor.....	84
SCM compiled script.....	85

Exact code - opcodes.....	85
Exact code - data types.....	85
Exact code - array, data next after primary variable.....	86
Exact code - array, type of multiplier and multiplier for index from secondary variable.....	86
Exact code - array example.....	86
Audio.....	87
Radio.....	87
List of radio stations.....	87
Memory.....	88
Information.....	88
How the process in memory works.....	88
Hex editors.....	89
NOP Operation:.....	89
Value types.....	89
Finding & patching memory addresses.....	90
Assembler short tips.....	93
Function addresses.....	94
Memory values.....	94
Uncategorized.....	94
Limits.....	94
Player.....	95
Time.....	95
Weather.....	95
CActor.....	96
Main struct.....	96
Actor struct.....	97
CVehicle - spawned vehicles.....	97
Cheats.....	97
SCM Opcodes.....	98
List of opcodes.....	98
Camera.....	98
Key.....	98
Math.....	98
NOP.....	100
Player.....	100

SCM Structure opcodes.....	100
Standard opcodes.....	100
0000 - NOP.....	100
0001 - WAIT.....	101
0002 - GOTO.....	102
0003 - SET_CAM_SHAKE.....	103
0004 - set_global_int_variable.....	103
0005 - set_global_float_variable.....	104
0006 - set_local_int_variable.....	105
0007 - set_local_float_variable.....	105
0008 - add_int_value_to_global_variable.....	106
0009 - add_float_value_to_global_variable.....	106
000A - add_int_value_to_local_variable.....	107
000B - add_float_value_to_local_variable.....	108
000C - subtract_int_value_from_global_variable.....	108
000D - subtract_float_value_from_global_variable.....	109
000E - subtract_int_value_from_local_variable.....	110
000F - subtract_float_value_from_local_variable.....	110
0010 - multiply_by_int_value_in_global_variable.....	111
0011 - multiply_by_float_value_in_global_variable.....	112
0012 - multiply_by_int_value_in_local_variable.....	112
0013 - multiply_by_float_value_in_local_variable.....	113
0014 - divide_by_int_value_in_global_variable.....	114
0015 - divide_by_float_value_in_global_variable.....	114
0016 - divide_by_int_value_in_local_variable.....	115
0017 - divide_by_float_value_in_local_variable.....	116
0018 - is_int_global_variable_greater_than_value.....	116
0019 - is_int_local_variable_greater_than_value.....	117
001A - is_int_value_greater_than_global_variable.....	118
001B - is_int_value_greater_than_local_variable.....	119
001C - is_int_global_variable_greater_than_global_variable.....	120
001D - is_int_local_variable_greater_than_local_variable.....	121
001E - is_int_global_variable_greater_than_local_variable.....	122
001F - is_int_local_variable_greater_than_global_variable.....	123
0020 - is_float_global_variable_greater_than_value.....	124

0021 - is_float_local_variable_greater_than_value.....	125
0022 - is_float_value_greater_than_global_variable.....	126
0023 - is_float_value_greater_than_local_variable.....	127
0024 - is_float_global_variable_greater_than_global_variable.....	128
0025 - is_float_local_variable_greater_than_local_variable.....	129
0026 - is_float_global_variable_greater_than_local_variable.....	130
0027 - is_float_local_variable_greater_than_global_variable.....	131
0028 - is_int_global_variable_greater_or_equal_to_value.....	132
0029 - is_int_local_variable_greater_or_equal_to_value.....	133
002A - is_int_value_greater_or_equal_to_global_variable.....	134
002B - is_int_value_greater_or_equal_to_local_variable.....	135
002C - is_int_global_variable_greater_or_equal_to_global_variable.....	136
002D - is_int_local_variable_greater_or_equal_to_local_variable.....	137
002E - is_int_global_variable_greater_or_equal_to_local_variable.....	138
002F - is_int_local_variable_greater_or_equal_to_global_variable.....	139
0030 - is_float_global_variable_greater_or_equal_to_value.....	140
0031 - is_float_local_variable_greater_or_equal_to_value.....	141
0032 - is_float_value_greater_or_equal_to_global_variable.....	142
0033 - is_float_value_greater_or_equal_to_local_variable.....	143
0034 - is_float_global_variable_greater_or_equal_to_global_variable.....	144
0034 - is_float_global_variable_greater_or_equal_to_global_variable.....	145
0035 - is_float_local_variable_greater_or_equal_to_local_variable.....	146
0036 - is_float_global_variable_greater_or_equal_to_local_variable.....	147
0037 - is_float_local_variable_greater_or_equal_to_global_variable.....	148
0038 - is_int_global_variable_equal_to_value.....	149
0039 - is_int_local_variable_equal_to_value.....	150
003A - is_int_global_variable_equal_to_global_variable.....	151
003B - is_int_local_variable_equal_to_local_variable.....	152
003C - is_int_global_variable_equal_to_local_variable.....	153
0042 - is_float_global_variable_equal_to_value.....	154
0043 - is_float_local_variable_equal_to_value.....	155
0044 - is_float_global_variable_equal_to_global_variable.....	156
0045 - is_float_local_variable_equal_to_local_variable.....	157
0046 - is_float_global_variable_equal_to_local_variable.....	158
004D - jump_if_false.....	159

004E - TERMINATE_THIS_SCRIPT.....	160
004F - START_NEW_SCRIPT_WITH_ARGS.....	161
0050 - gosub.....	163
0051 - return.....	163
0052 - NOP_floats.....	165
0053 - CREATE_PLAYER.....	166
0058 - add_int_global_variable_to_global_variable.....	167
0059 - add_float_global_variable_to_global_variable.....	168
005A - add_int_local_variable_to_local_variable.....	169
CLE03 opcodes.....	170
0A8C - write_memory.....	170
0AB0 - virtual_key_pressed.....	170
ASCII table.....	171
Terminology.....	179
References.....	179

General

There are patches for GTA San Andreas. In this documentation it is tried to explain things for:

1. GTA San Andreas v1.0 [US] HOODLUM No-CD Fixed EXE:
EXE size: **14 383 616** bytes
EXE MD5: **170b3a9108687b26da2d8901c6948a18**
EXE SHA-1: **185b73fbceaa05d66452691fc0d15c8d61b92a7e**
2. GTA: San Andreas v1.01 [EURO] No-CD/Fixed EXE:
EXE size: **15 806 464** bytes
EXE MD5: **25405921d1c47747fd01fd0bfe0a05ae**
EXE SHA-1: **de9ebfe4943d1d1888b8adabfef2e7d4fa4f0943**

Limitations

GTA San Andreas imposes specific limits on certain types of object instances and files. Exceeding some of these limits can make the game crash. Or the game may skip the extra data and load only as much as there is place in memory for. While several limits can be overcome using SA limit adjuster, many are still yet to be hacked.

This table lists short names for the limits, long descriptions, their default values and whether or not they can be hacked with existing plugins.

<i>Short name</i>	<i>Description</i>	<i>Default value of limit</i>	<i>How to overcome</i>
<i>Map limits</i>			
IPL Instances	The Number of IPL Instances allowed in the game (inst). Every placed objects increases of IPL instances e.g. buildings.	13000	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Dummies	NOTE: Increase to increase IPL Limit	2500	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required

PtrNode Singles	-	70000	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
PtrNode Doubles	NOTE: Increase to increase IPL Limit	3200	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
In IMG, IPL Files	The number of allowed binary IPL files in IMG archives.	256	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Stunt Jumps	The number of stunt jumps allowed in the game.	256	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Collision models	The number of all collisions. COL file can contain more than one collision model.	10150	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Collision files	The number which describes how many COL files can exist in all of IMG archives. There are 251 .col files in unmodified IMG archives.	254	-
QuadTreeNodes	-	400	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required

QuadTreeNodes2	-	40	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
MatDataPool	-	4096	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
AtmDataPool	-	1024	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Spawned vehicles in pool	Limits determines how many vehicles can be spawned and exist in pool at the same time, "now".	110	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Vehicle IDs	How many vehicles can be declared? Each of vehicles is declared on its own ID in IDE file, by default it is vehicles.ide	212	Use SA Limit Adjuster or Alexander Blade More Vehicles plugin. Vehicle Audio Loader for sounds of new vehicles gta_sa.exe 1.0 required
Spawned peds in pool	Limits determines how many peds can be spawned and exist in pool at the same time, "now".	140	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Ped IDs	How many ped can be declared? Each of peds is declared on its own ID in IDE file, by default it is peds.ide	278	Use SA Limit Adjuster, gta_sa.exe 1.0 required

Polygons	The maximum number of Polygons that a model can have	5000	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
EntryInfoNodes	Number of pickupable items you can create via SCM Scripts	500	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Objects	The Number of Dynamic Objects creatable	350	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Tasks	-	500	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Events	-	200	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Point Routes	-	64	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Patrol Routes	-	32	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required

Node Routes	-	64	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Task Allocators	-	64	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
Ped Attractors	-	64	Use SA Limit Adjuster or Alexander Blade SA Limit Adjuster gta_sa.exe 1.0 required
cargrp.dat	Cargrp.dat assigns vehicles to groups which are assigned to zones by SCM script.	34 groups, 23 vehicles per zone. Last 4 groups are reserved for cheats.	-
pedgrp.dat	Pedgrp.dat assigns vehicles to groups which are assigned to zones by SCM script.	57 groups, 21 peds per zone.	-
Entry Exits	Maximum number of Enex's able to be defined in IPL's	400 entry exits	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Streaming Memory	Amount of memory for loading textures around camera. When too many high-res textures are added to the game they disappear because there isn't enough memory. Increasing this limit solves texture loading problem.	25600000 B = 25000 KB = 24.4140625 MB	Use SA Limit Adjuster or Alexander Blade Stream Memory Fix, his 2.0 version increases streaming for vehicles and crashes game. 1.0 is good; gta_sa.exe 1.0 required

Timed Objects	The number of tobj's that can be defined in IDE files	227 objects	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Water Planes	The maximum number of water planes defined in water.dat, they are	1021 water planes	Use SA Limit Adjuster, gta_sa.exe 1.0 required
IDE objects	The maximum number of declared objects in .ide files by objs keyword.	14000 objects	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Carmods.dat	Carmods.dat contains list of cars and the mods that can be added to these cars. A limit describes how many cars can occur with parts in carmods.dat	70 tunable cars	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Carcols.dat	The max number of colours in colour table defined in carcols.dat	128 colours	Use SA Limit Adjuster, gta_sa.exe 1.0 required
SCM Mission size	The max number of bytes one mission can have in main.scm	69000 bytes = 67.4 KB	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Mission local variables	Number of available local variables in missions	1024 (0@ - 1023@)	-
Thread local variables	Number of available local variables in threads	34; 0@ - 31@ normal local variables 32@ - 33@ INT millisecond timers growing up with time	-
SCM Threads	Number of created threads by 004F or 00D7 opcode in main.scm	96 threads	Use SA Limit Adjuster, gta_sa.exe 1.0 required

Mission Cleanup List	The max number of vehicle handles which can be added to cleanup list during mission. These vehicles will disappear after an 00D8 opcode is executed.	75 vehicle handles	-
IMG Files	The max number of loaded IMG files	8 IMG files: 5 of them can be declared in basic map file 3 of them are hard-coded	
IMG Headers	IMG Header Allocation	5000	Use SA Limit Adjuster, gta_sa.exe 1.0 required
The area defining the mapping bounds	Probably the maximal coordinates where could be an object placed and work correctly.	12000.0 coordinate points	Use SA Limit Adjuster, gta_sa.exe 1.0 required
LOD Distance	Game loads higher textures and models around up to specific distance. After that distance textures aren't fully loaded.	300.0 coordinate points	Use SA Limit Adjuster, gta_sa.exe 1.0 required
Markers	Number of active markers created by SCM script.	175 markers	Use SA Limit Adjuster, buggy gta_sa.exe 1.0 required
Garages	The max number of garages added in IDE files within grge section	50 garages	Use SA Limit Adjuster, buggy gta_sa.exe 1.0 required

Pickups	The max number of active pickups created by SCM script. Weapons on ground, save icons – they are examples of pickups	620 pickups	Use SA Limit Adjuster, buggy gta_sa.exe 1.0 required
Players	The max number of players, all of them move on key presses together, but camera is on recently created player.	8 players	Use SA Limit Adjuster, buggy gta_sa.exe 1.0 required
Weapon IDs	Number of weapons which are defined in gta_sa.exe and use handling in weapon.dat	47 weapons	-
Towns	Number of towns which have individual weathers, police cars and peds.	3 towns	-
Map coordinates	Max coordinates of area displayed on map in menu or radar.	From -3000.0 to 3000.0 units. 6000 units	-
Radio stations	Number of existing radios	12 radio stations	-

Primary loaded files

At the beginning GTA San Andreas exe loads primary files which point to another files. These files cannot be removed because the game expects their existence. They can be divided into following parts:

<u>Menu, before loading or starting new game</u>	
<i>Movies before loading</i>	
movies\Logo.mpg	Video containing Rockstar Games logo.
models\GTAtitles.mpg	Video presenting GTA San Andreas
models\txd\LOADSCS.txd	Splash screens, EAX, Nvidia logos. Textures displayed on loading the game.
Menu textures	
models\fronten_pc.txd	Mouse cursor in menu & crosshair texture
models\fronten1.txd	Radio station logos
models\fronten2.txd	Textures drawn on corners in menu
models\fronten3.txd	Seems like unused textures being on purpose to be drawn on top and right corner of menu.
models\pcbtns.txd	Right, left, up, down arrows
<i>Collision file</i>	
models\coll\peds.col	<i>Ped collisions, it is not standard collision file</i>
<i>Audio files</i>	
audio\CONFIG\BankLkup.dat audio\CONFIG\BankSlot.dat audio\CONFIG\EventVol.dat audio\CONFIG\PakFiles.dat audio\CONFIG\StrmPaks.dat audio\CONFIG\TrakLkup.dat	Configuration of audio files, following contain names, positions and sizes of tracks, sounds in particular. It is very simplified information. PakFiles.dat contains names of audio archives from SFX directory. StrmPaks.dat stores names of audio archives from STREAMS directory. None of archives can be removed; only name can be changed because GTA SA refers to them by IDs.
audio\SFX*	Depending on PakFiles.dat an archives from this directory are loaded. Files contain sounds, voices, engine effects and so forth.
audio\streams*	Radio stations, police, adverts and ambience. Archive names depend on StrmPaks.dat
<i>Decision files</i>	

data\Decision\PedEvent.txt	Decision ped events. Loaded after loading screen is displayed and before a menu arises.
data\Decision\Allowed*	Decision files are used only by decision opcodes. Removing them doesn't make crash the game. Decision opcodes use them so that they should be kept.
<i>Fonts</i>	
data\fonts.dat	Contains ASCII character table and positions to display chars
models\fonts.txd	Characters as transparent alpha textures.
<i>Language files</i>	
text\american.gxt text\french.gxt text\german.gxt text\italian.gxt text\spanish.gxt	Depending on current language and their names in exe or memory one of .gxt files is loaded. GXT file consists of short GXT names (hashed) and texts. Not exactly GXT names, just hashes for which game must know GXT name to find GXT entry with proper hash.

<u>On loading the game</u>	
File	Description
<i>Game scripting, missions</i>	
data\script\main.scm	Main.scm is mission scripting file loaded when standard game is selected. It consists of declared objects, scripts and lot of opcodes. Things to do in missions are programmed and compiled to this file. It is binary file. Compiled main.scm with included external scripts requires .scm files placed in any of loaded IMG archives. Usually data\script\script.img is defined in gta.dat for this purpose.
<i>Routes, paths</i>	
data\Paths\ROADBLOX.DAT	Contains information about where to put road blocks during police the pursuit.
data\Paths\tracks.dat data\Paths\tracks2.dat data\Paths\tracks3.dat data\Paths\tracks4.dat	Following files contain train routes. These are text files and comprise of coordinates and angles.
<i>Most primary files referring to another game files.</i>	

data\default.dat data\gta.dat	Both files have the same format. They are most primary files which refer to load external IMG, COL, IDE, IPL, ZON, TXD files. Mainly used for loading IPL, IDE and IPL files like default.ide, peds.ide, vehicles.ide. IDE files should be declared before IPL in these two files
<i>Hardcoded IMG archives to load</i>	
anim\cuts.img models\gta3.img models\gta_int.img models\player.img	<p>These are IMG archives those names are hardcoded in EXE to load. IMG archives contain the game files, usually:</p> <ul style="list-style-type: none"> • .dff models • .txd texture archives • .col collision sets • .ifp animations for peds or objects • .cut cutscene text data • .dat cutscene camera movements <p>Note that cuts.img is only opened when start cutscene is executed and GTA San Andreas is playing cutscene.</p>
<i>Map related files</i>	
data\furnitur.dat	Interior furniture groups data. This file is actually unused and can be left blank, but it must not be deleted, since the game tries to parse it. It originally contains information about furniture which was planned to be used for automatically generating different furniture in interiors. However, this file, and the interior models/collision files are not used by the game engine, and other required files are missing.
data\object.dat	Contains some object properties, for example it is used for destroyable objects. Object is identified by his model name and had to be declared in IDE file.
data\plants.dat	It contains information of plants. Names are hard-coded in EXE file and were compiled from ColPoint.h

data\procobj.dat	Dynamic objects are defined in the object.dat file. They must be defined in an IDE file first. With additionally defining them in the object.dat file they will have special abilities. Dynamic objects are used to create objects which are able to interact with their environment. For example they are able to explode or open when the player approaches them (garage doors).
data\surface.dat	Contains multiplies of surfaces for WheelBase (Rubber, Hard, Road, Loose, Sand, Wet).
data\surfaud.dat	It assigns audio to a collision when objects interact with the surfaces. The interactions include the footsteps peds make and vehicles sliding on the surface.
data\surfinfo.dat	Contains information of objects how to interact visually. For example sand objects when car tires sink in and can get bogged down.
data\water.dat	The water.dat file contains information about the water planes inside a boundary, which normally has the same size as the map boundary
<i>Game apperance</i>	
data\timecyc.dat	The timecyc.dat data file contains the most important settings about the game's appearance. Basically it stores important information like colors and lighting for different hardcoded weather situations for each time of day. The settings themselves are constant and build a cycle around the whole day. So the exact name of the setting is time cycle.
<i>Player related files</i>	
data\ar_stats.dat	Stores the stats reaction variables
data\clothes.dat	File listing cutscene replacements for clothes models and rules about what clothes models can be used with what. It contains rules to change clothes if some other clothes don't match or cutscene is loaded.
data\shopping.dat	File which has every object's price in it. The file is broken into sections by type of object e.g. car mod, furniture. Tunable parts, headdresses need to have prices.

data\statdisp.dat	Selects when to show messages based on stat values in the game. When stat of player is getting lower or higher than certain value, the game can show message, for example “
<i>Ped & Player and vehicle related files</i>	
data\melee.dat	Contains information on melee attacks, including damage and animation
data\pedstats.dat	Ped stats which are used for declared peds.
<i>Ped only related files</i>	
data\animgrp.dat	File describes the anim association groups. There are walk cycle groups described. These walk cycles can be applied to peds in IDE files.
anim\ped.ifp	Animations which don't need to be loaded by opcodes.
data\ped.dat	# Acquaintance options: # - Hate # - Dislike # - Like # - Respect There is specified group for every ped. This file controls how groups react to peds from other groups.
data\pedgrp.dat	Contains a list of ped groups for each group type. Each ped group should contain 32 ped type names. Please don't use 'male01' in any of the pedgroups
data\weapon.dat	The file contains information about the weapons properties and settings. The weapons themselves are defined inside the WEAP section in IDE file, usually default.ide located in the same folder. IDs of weapons and sounds are hardcoded in executable.
<i>Ped and vehicle related file</i>	
data\popcycle.dat	Fore each type of zone (Business, Countryside etc.) we have a that controls the ped densities.
<i>Vehicle related files</i>	
data\cargrp.dat	Zones are defined .zon file as coordinate corners. Each zone can be associated to car group by SCM script. Vehicles are assigned to specific groups in cargrp.dat. Cars drive on paths depending on car groups for current coordinates.

data\carcols.dat	This file contains all the info about the car colours. There are two sections, col which contains the colour table and car which contains the possible indices into the colour table for each car.
data\carmods.dat	List of cars and the mods that can be added to these cars. Each of tunable cars must be listed in this file.
data\handling.cfg	The handling.cfg file is a text data file format which sets many performance and behaviour values for each vehicle. The file can be opened and edited with any text editor.
<i>Common vehicle textures</i>	
models\generic\vehicle.txd	
<i>Grass</i>	
models\grass*.*	Grass models. Any DFF file can be put to grass directory and used in IPL files.
<i>Effects</i>	
models\effects.fxp	Text file containing an animations of moving textures to simulate fuzzy effects.
models\effectsPC.txd	Effect textures e.g. fire, smoke and so on.
models\particle.txd	Contains world particle like clouds, water, shadow, blood texture.

This list covers files whose paths are hardcoded in executable. That`s why they are called primary. Some of them (default.dat & gta.dat) point to another files and some need settings in other files.

Two of most important files are data\gta.dat and data\default.dat which point to external map files (IDE and IPL, COL and so on). They control what additional IMG archives to load.

The files the game loads for world like models, textures and collisions are stored in binary format in IMG archives. They can be created using 3dsMax and contain geometry of objects.

A game sequence of working

Game is run from .exe file and then load screens including a video are displayed. After the user chooses to load or start a new game, whole configuration and files are parsed, processed or opened.

Finally the game is loaded and user is able to control the player or exit to paused menu.

While game is not paused and player is playing, application is programmed to constantly some of procedures between each frame. Moving to a next frame is called "passage" here. During the passage we can distinguish following actions:

- Process world tasks
- Process world physics
- Process keys
- Process SCM scripts, scripting. Threads are processed in descending order.
- Render textures up to this frame and remove their information, next request of rendering textures by SCM script is necessary for each frame to keep the textures being displayed.

ASCII table

ASCII control characters (character code 0-31)					
The first 32 characters in the ASCII-table are unprintable control codes and are used to control peripherals such as printers.					
Dec	Hex	Oct	Binary	Symbol	Description
0	0x00	000	0000000 0	NUL	Null char
1	0x01	001	0000000 1	SOH	Start of Heading
2	0x02	002	0000001 0	STX	Start of Text
3	0x03	003	0000001 1	ETX	End of Text
4	0x04	004	0000010 0	EOT	End of Transmission
5	0x05	005	0000010 1	ENQ	Enquiry
6	0x06	006	0000011 0	ACK	Acknowledgment
7	0x07	007	0000011 1	BEL	Bell
8	0x08	010	0000100 0	BS	Back Space
9	0x09	011	0000100 1	HT	Horizontal Tab
10	0x0A	012	0000101 0	LF	Line Feed
11	0x0B	013	0000101 1	VT	Vertical Tab
12	0x0C	014	0000110 0	FF	Form Feed
13	0x0D	015	0000110 1	CR	Carriage Return
14	0x0E	016	0000111 0	SO	Shift Out / X-On
15	0x0F	017	0000111 1	SI	Shift In / X-Off
16	0x10	020	0001000 0	DLE	Data Line Escape
17	0x11	021	0001000 1	DC1	Device Control 1 (oft. XON)
18	0x12	022	0001001 0	DC2	Device Control 2

19	0x13	023	0001001 1	DC3	Device Control 3 (oft. XOFF)
20	0x14	024	0001010 0	DC4	Device Control 4
21	0x15	025	0001010 1	NAK	Negative Acknowledgeme nt
22	0x16	026	0001011 0	SYN	Synchronous Idle
23	0x17	027	0001011 1	ETB	End of Transmit Block
24	0x18	030	0001100 0	CAN	Cancel
25	0x19	031	0001100 1	EM	End of Medium
26	0x1A	032	0001101 0	SUB	Substitute
27	0x1B	033	0001101 1	ESC	Escape
28	0x1C	034	0001110 0	FS	File Separator
29	0x1D	035	0001110 1	GS	Group Separator
30	0x1E	036	0001111 0	RS	Record Separator
31	0x1F	037	0001111 1	US	Unit Separator

ASCII printable characters (character code 32-127)

Codes 32-127 are common for all the different variations of the ASCII table, they are called printable characters, represent letters, digits, punctuation marks, and a few miscellaneous symbols. You will find almost every character on your keyboard. Character 127 represents the command DEL.

<u>Dec</u>	<u>Hex</u>	<u>Oct</u>	<u>Binary</u>	<u>Symbol</u>	<u>Description</u>
32	0x20	040	00100000		Space
33	0x21	041	00100001	!	Exclamation mark
34	0x22	042	00100010	"	Double quotes (or speech marks)
35	0x23	043	00100011	#	Number
36	0x24	044	00100100	\$	Dollar
37	0x25	045	00100101	%	Procenttecken
38	0x26	046	00100110	&	Ampersand
39	0x27	047	00100111	'	Single quote
40	0x28	050	00101000	(Open parenthesis (or open bracket)
41	0x29	051	00101001)	Close parenthesis (or close bracket)
42	0x2A	052	00101010	*	Asterisk
43	0x2B	053	00101011	+	Plus
44	0x2C	054	00101100	,	Comma

45	0x2D	055	00101101	-	Hyphen
46	0x2E	056	00101110	.	Period, dot or full stop
47	0x2F	057	00101111	/	Slash or divide
48	0x30	060	00110000	0	Zero
49	0x31	061	00110001	1	One
50	0x32	062	00110010	2	Two
51	0x33	063	00110011	3	Three
52	0x34	064	00110100	4	Four
53	0x35	065	00110101	5	Five
54	0x36	066	00110110	6	Six
55	0x37	067	00110111	7	Seven
56	0x38	070	00111000	8	Eight
57	0x39	071	00111001	9	Nine
58	0x3A	072	00111010	:	Colon
59	0x3B	073	00111011	;	Semicolon
60	0x3C	074	00111100	<	Less than (or open angled bracket)
61	0x3D	075	00111101	=	Equals
62	0x3E	076	00111110	>	Greater than (or close angled bracket)
63	0x3F	077	00111111	?	Question mark
64	0x40	100	01000000	@	At symbol
65	0x41	101	01000001	A	Uppercase A
66	0x42	102	01000010	B	Uppercase B
67	0x43	103	01000011	C	Uppercase C
68	0x44	104	01000100	D	Uppercase D
69	0x45	105	01000101	E	Uppercase E
70	0x46	106	01000110	F	Uppercase F
71	0x47	107	01000111	G	Uppercase G
72	0x48	110	01001000	H	Uppercase H
73	0x49	111	01001001	I	Uppercase I
74	0x4A	112	01001010	J	Uppercase J
75	0x4B	113	01001011	K	Uppercase K
76	0x4C	114	01001100	L	Uppercase L
77	0x4D	115	01001101	M	Uppercase M
78	0x4E	116	01001110	N	Uppercase N
79	0x4F	117	01001111	O	Uppercase O
80	0x50	120	01010000	P	Uppercase P
81	0x51	121	01010001	Q	Uppercase Q
82	0x52	122	01010010	R	Uppercase R
83	0x53	123	01010011	S	Uppercase S
84	0x54	124	01010100	T	Uppercase T
85	0x55	125	01010101	U	Uppercase U
86	0x56	126	01010110	V	Uppercase V
87	0x57	127	01010111	W	Uppercase W
88	0x58	130	01011000	X	Uppercase X
89	0x59	131	01011001	Y	Uppercase Y
90	0x5A	132	01011010	Z	Uppercase Z
91	0x5B	133	01011011	[Opening bracket

92	0x5C	134	01011100	\	Backslash
93	0x5D	135	01011101]	Closing bracket
94	0x5E	136	01011110	^	Caret - circumflex
95	0x5F	137	01011111	_	Underscore
96	0x60	140	01100000	`	Grave accent
97	0x61	141	01100001	a	Lowercase a
98	0x62	142	01100010	b	Lowercase b
99	0x63	143	01100011	c	Lowercase c
100	0x64	144	01100100	d	Lowercase d
101	0x65	145	01100101	e	Lowercase e
102	0x66	146	01100110	f	Lowercase f
103	0x67	147	01100111	g	Lowercase g
104	0x68	150	01101000	h	Lowercase h
105	0x69	151	01101001	i	Lowercase i
106	0x6A	152	01101010	j	Lowercase j
107	0x6B	153	01101011	k	Lowercase k
108	0x6C	154	01101100	l	Lowercase l
109	0x6D	155	01101101	m	Lowercase m
110	0x6E	156	01101110	n	Lowercase n
111	0x6F	157	01101111	o	Lowercase o
112	0x70	160	01110000	p	Lowercase p
113	0x71	161	01110001	q	Lowercase q
114	0x72	162	01110010	r	Lowercase r
115	0x73	163	01110011	s	Lowercase s
116	0x74	164	01110100	t	Lowercase t
117	0x75	165	01110101	u	Lowercase u
118	0x76	166	01110110	v	Lowercase v
119	0x77	167	01110111	w	Lowercase w
120	0x78	170	01111000	x	Lowercase x
121	0x79	171	01111001	y	Lowercase y
122	0x7A	172	01111010	z	Lowercase z
123	0x7B	173	01111011	{	Opening brace
124	0x7C	174	01111100		Vertical bar
125	0x7D	175	01111101	}	Closing brace
126	0x7E	176	01111110		Equivalency sign - tilde
127	0x7F	177	01111111		Delete

The extended ASCII codes (character code 128-255)

There are several different variations of the 8-bit ASCII table. The table below is according to ISO 8859-1, also called ISO Latin-1. Codes 129-159 contain the Microsoft® Windows Latin-1 extended characters.

128	0x80	200	10000000	€	Euro sign
129	0x81	201	10000001		
130	0x82	202	10000010	,	Single low-9 quotation mark
131	0x83	203	10000011	ƒ	Latin small letter f with hook
132	0x84	204	10000100	„	Double low-9 quotation mark
133	0x85	205	10000101	...	Horizontal ellipsis

134	0x86	206	10000110	†	Dagger
135	0x87	207	10000111	‡	Double dagger
136	0x88	210	10001000	^	Modifier letter circumflex accent
137	0x89	211	10001001	‰	Per mille sign
138	0x8A	212	10001010	Š	Latin capital letter S with caron
139	0x8B	213	10001011	◁	Single left-pointing angle quotation
140	0x8C	214	10001100	Œ	Latin capital ligature OE
141	0x8D	215	10001101		
142	0x8E	216	10001110	Ž	Latin capital letter Z with caron
143	0x8F	217	10001111		
144	0x90	220	10010000		
145	0x91	221	10010001	‘	Left single quotation mark
146	0x92	222	10010010	’	Right single quotation mark
147	0x93	223	10010011	“	Left double quotation mark
148	0x94	224	10010100	”	Right double quotation mark
149	0x95	225	10010101	•	Bullet
150	0x96	226	10010110	–	En dash
151	0x97	227	10010111	—	Em dash
152	0x98	230	10011000	~	Small tilde
153	0x99	231	10011001	™	Trade mark sign
154	0x9A	232	10011010	š	Latin small letter S with caron
155	0x9B	233	10011011	▷	Single right-pointing angle quotation mark
156	0x9C	234	10011100	œ	Latin small ligature oe
157	0x9D	235	10011101		
158	0x9E	236	10011110	ž	Latin small letter z with caron
159	0x9F	237	10011111	ÿ	Latin capital letter Y with diaeresis
160	0xA0	240	10100000		Non-breaking space
161	0xA1	241	10100001	¡	Inverted exclamation mark
162	0xA2	242	10100010	¢	Cent sign
163	0xA3	243	10100011	£	Pound sign

164	0xA4	244	10100100	¤	Currency sign
165	0xA5	245	10100101	¥	Yen sign
166	0xA6	246	10100110		Pipe, Broken vertical bar
167	0xA7	247	10100111	§	Section sign
168	0xA8	250	10101000	¨	Spacing diaeresis - umlaut
169	0xA9	251	10101001	©	Copyright sign
170	0xAA	252	10101010	ª	Feminine ordinal indicator
171	0xAB	253	10101011	«	Left double angle quotes
172	0xAC	254	10101100	¬	Not sign
173	0xAD	255	10101101		Soft hyphen
174	0xAE	256	10101110	®	Registered trade mark sign
175	0xAF	257	10101111	¯	Spacing macron - overline
176	0xB0	260	10110000	°	Degree sign
177	0xB1	261	10110001	±	Plus-or-minus sign
178	0xB2	262	10110010	²	Superscript two - squared
179	0xB3	263	10110011	³	Superscript three - cubed
180	0xB4	264	10110100	´	Acute accent - spacing acute
181	0xB5	265	10110101	µ	Micro sign
182	0xB6	266	10110110	¶	Pilcrow sign - paragraph sign
183	0xB7	267	10110111	·	Middle dot - Georgian comma
184	0xB8	270	10111000	¸	Spacing cedilla
185	0xB9	271	10111001	¹	Superscript one
186	0xBA	272	10111010	º	Masculine ordinal indicator
187	0xBB	273	10111011	»	Right double angle quotes
188	0xBC	274	10111100	¼	Fraction one quarter
189	0xBD	275	10111101	½	Fraction one half
190	0xBE	276	10111110	¾	Fraction three quarters
191	0xBF	277	10111111	¿	Inverted question mark
192	0xC0	300	11000000	À	Latin capital letter A with grave
193	0xC1	301	11000001	Á	Latin capital letter A with acute

194	0xC2	302	11000010	Â	Latin capital letter A with circumflex
195	0xC3	303	11000011	Ã	Latin capital letter A with tilde
196	0xC4	304	11000100	Ä	Latin capital letter A with diaeresis
197	0xC5	305	11000101	Å	Latin capital letter A with ring above
198	0xC6	306	11000110	Æ	Latin capital letter AE
199	0xC7	307	11000111	Ç	Latin capital letter C with cedilla
200	0xC8	310	11001000	È	Latin capital letter E with grave
201	0xC9	311	11001001	É	Latin capital letter E with acute
202	0xCA	312	11001010	Ê	Latin capital letter E with circumflex
203	0xCB	313	11001011	Ë	Latin capital letter E with diaeresis
204	0xCC	314	11001100	Ì	Latin capital letter I with grave
205	0xCD	315	11001101	Í	Latin capital letter I with acute
206	0xCE	316	11001110	Î	Latin capital letter I with circumflex
207	0xCF	317	11001111	Ï	Latin capital letter I with diaeresis
208	0xD0	320	11010000	Ð	Latin capital letter ETH
209	0xD1	321	11010001	Ñ	Latin capital letter N with tilde
210	0xD2	322	11010010	Ò	Latin capital letter O with grave
211	0xD3	323	11010011	Ó	Latin capital letter O with acute

212	0xD4	324	11010100	Ô	Latin capital letter O with circumflex
213	0xD5	325	11010101	Õ	Latin capital letter O with tilde
214	0xD6	326	11010110	Ö	Latin capital letter O with diaeresis
215	0xD7	327	11010111	×	Multiplication sign
216	0xD8	330	11011000	Ø	Latin capital letter O with slash
217	0xD9	331	11011001	Ù	Latin capital letter U with grave
218	0xDA	332	11011010	Ú	Latin capital letter U with acute
219	0xDB	333	11011011	Û	Latin capital letter U with circumflex
220	0xDC	334	11011100	Ü	Latin capital letter U with diaeresis
221	0xDD	335	11011101	Ý	Latin capital letter Y with acute
222	0xDE	336	11011110	Þ	Latin capital letter THORN
223	0xDF	337	11011111	ß	Latin small letter sharp s - ess-zed
224	0xE0	340	11100000	à	Latin small letter a with grave
225	0xE1	341	11100001	á	Latin small letter a with acute
226	0xE2	342	11100010	â	Latin small letter a with circumflex
227	0xE3	343	11100011	ã	Latin small letter a with tilde
228	0xE4	344	11100100	ä	Latin small letter a with diaeresis
229	0xE5	345	11100101	å	Latin small letter a with ring above
230	0xE6	346	11100110	æ	Latin small letter æ
231	0xE7	347	11100111	ç	Latin small letter c with cedilla
232	0xE8	350	11101000	è	Latin small letter e with grave

233	0xE9	351	11101001	é	Latin small letter e with acute
234	0xEA	352	11101001	ê	Latin small letter e with circumflex
235	0xEB	353	11101011	ë	Latin small letter e with diaeresis
236	0xEC	354	11101100	ì	Latin small letter i with grave
237	0xED	355	11101101	í	Latin small letter i with acute
238	0xEE	356	11101110	î	Latin small letter i with circumflex
239	0xEF	357	11101111	ï	Latin small letter i with diaeresis
240	0xF0	360	11110000	ð	Latin small letter eth
241	0xF1	361	11110001	ñ	Latin small letter n with tilde
242	0xF2	362	11110010	ò	Latin small letter o with grave
243	0xF3	363	11110011	ó	Latin small letter o with acute
244	0xF4	364	11110100	ô	Latin small letter o with circumflex
245	0xF5	365	11110101	õ	Latin small letter o with tilde
246	0xF6	366	11110110	ö	Latin small letter o with diaeresis
247	0xF7	367	11110111	÷	Division sign
248	0xF8	370	11111000	ø	Latin small letter o with slash
249	0xF9	371	11111001	ù	Latin small letter u with grave
250	0xFA	372	11111010	ú	Latin small letter u with acute
251	0xFB	373	11111011	û	Latin small letter u with circumflex
252	0xFC	374	11111100	ü	Latin small letter u with diaeresis
253	0xFD	375	11111101	ý	Latin small letter y with acute
254	0xFE	376	11111110	þ	Latin small letter thorn
255	0xFF	377	11111111	ÿ	Latin small letter y with diaeresis

Virtual key codes

The following table shows the symbolic constant names, hexadecimal values, and mouse or keyboard equivalents for the virtual-key codes used by the system. The codes are listed in numeric order.

Costant	Description	Decimal value	Hexadecimal value
VK_LBUTTON	Left mouse button	1	0x01
VK_RBUTTON	Right mouse button	2	0x02
VK_CANCEL	Control-break processing	3	0x03
VK_MBUTTON	Middle mouse button (three-button mouse)	4	0x04
VK_XBUTTON1	X1 mouse button	5	0x05
VK_XBUTTON2	X2 mouse button	6	0x06
-	Undefined	7	0x07
VK_BACK	BACKSPACE key	8	0x08
VK_TAB	TAB key	9	0x09
-	Reserved	10-11	0x0A-0x0B
VK_CLEAR	CLEAR key	12	0x0C
VK_RETURN	ENTER key	13	0x0D
-	Undefined	14-15	0x0E-0x0F
VK_SHIFT	SHIFT key	16	0x10
VK_CONTROL	CTRL key	17	0x11
VK_MENU	ALT key	18	0x12
VK_PAUSE	PAUSE key	19	0x13
VK_CAPITAL	CAPS LOCK key	20	0x14
VK_KANA	IME Kana mode	21	0x15
VK_HANGUEL	IME Hanguel mode (maintained for compatibility; use VK_HANGUL)	21	0x15
VK_HANGUL	IME Hangul mode	21	0x15
-	Undefined	22	0x16
VK_JUNJA	IME Junja mode	23	0x17
VK_FINAL	IME final mode	24	0x18
VK_HANJA	IME Hanja mode	25	0x19
VK_KANJI	IME Kanji mode	25	0x19
-	Undefined	26	0x1A
VK_ESCAPE	ESC key	27	0x1B
VK_CONVERT	IME convert	28	0x1C
VK_NONCONVERT	IME nonconvert	29	0x1D
VK_ACCEPT	IME accept	30	0x1E
VK_MODECHANGE	IME mode change request	31	0x1F
VK_SPACE	SPACEBAR	32	0x20
VK_PRIOR	PAGE UP key	33	0x21
VK_NEXT	PAGE DOWN key	34	0x22

VK_END	END key	35	0x23
VK_HOME	HOME key	36	0x24
VK_LEFT	LEFT ARROW key	37	0x25
VK_UP	UP ARROW key	38	0x26
VK_RIGHT	RIGHT ARROW key	39	0x27
VK_DOWN	DOWN ARROW key	40	0x28
VK_SELECT	SELECT key	41	0x29
VK_PRINT	PRINT key	42	0x2A
VK_EXECUTE	EXECUTE key	43	0x2B
VK_SNAPSHOT	PRINT SCREEN key	44	0x2C
VK_INSERT	INS key	45	0x2D
VK_DELETE	DEL key	46	0x2E
VK_HELP	HELP key	47	0x2F
VK_KEY_0	0 key	48	0x30
VK_KEY_1	1 key	49	0x31
VK_KEY_2	2 key	50	0x32
VK_KEY_3	3 key	51	0x33
VK_KEY_4	4 key	52	0x34
VK_KEY_5	5 key	53	0x35
VK_KEY_6	6 key	54	0x36
VK_KEY_7	7 key	55	0x37
VK_KEY_8	8 key	56	0x38
VK_KEY_9	9 key	57	0x39
-	Undefined	58-64	0x3A-40
VK_KEY_A	A key	65	0x41
VK_KEY_B	B key	66	0x42
VK_KEY_C	C key	67	0x43
VK_KEY_D	D key	68	0x44
VK_KEY_E	E key	69	0x45
VK_KEY_F	F key	70	0x46
VK_KEY_G	G key	71	0x47
VK_KEY_H	H key	72	0x48
VK_KEY_I	I key	73	0x49
VK_KEY_J	J key	74	0x4A
VK_KEY_K	K key	75	0x4B
VK_KEY_L	L key	76	0x4C
VK_KEY_M	M key	77	0x4D
VK_KEY_N	N key	78	0x4E
VK_KEY_O	O key	79	0x4F
VK_KEY_P	P key	80	0x50
VK_KEY_Q	Q key	81	0x51
VK_KEY_R	R key	82	0x52
VK_KEY_S	S key	83	0x53
VK_KEY_T	T	84	0x54
VK_KEY_U	U key	85	0x55
VK_KEY_V	V key	86	0x56
VK_KEY_W	W key	87	0x57
VK_KEY_X	X key	88	0x58
VK_KEY_Y	Y key	89	0x59

VK_KEY_Z	Z key	90	0x5A
VK_LWIN	Left Windows key (Natural keyboard)	91	0x5B
VK_RWIN	Right Windows key (Natural keyboard)	92	0x5C
VK_APPS	Applications key (Natural keyboard)	93	0x5D
-	Reserved	94	0x5E
VK_SLEEP	Computer Sleep key	95	0x5F
VK_NUMPAD0	Numeric keypad 0 key	96	0x60
VK_NUMPAD1	Numeric keypad 1 key	97	0x61
VK_NUMPAD2	Numeric keypad 2 key	98	0x62
VK_NUMPAD3	Numeric keypad 3 key	99	0x63
VK_NUMPAD4	Numeric keypad 4 key	100	0x64
VK_NUMPAD5	Numeric keypad 5 key	101	0x65
VK_NUMPAD6	Numeric keypad 6 key	102	0x66
VK_NUMPAD7	Numeric keypad 7 key	103	0x67
VK_NUMPAD8	Numeric keypad 8 key	104	0x68
VK_NUMPAD9	Numeric keypad 9 key	105	0x69
VK_MULTIPLY	Multiply key	106	0x6A
VK_ADD	Add key	107	0x6B
VK_SEPARATOR	Separator key	108	0x6C
VK_SUBTRACT	Subtract key	109	0x6D
VK_DECIMAL	Decimal key	110	0x6E
VK_DIVIDE	Divide key	111	0x6F
VK_F1	F1 key	112	0x70
VK_F2	F2 key	113	0x71
VK_F3	F3 key	114	0x72
VK_F4	F4 key	115	0x73
VK_F5	F5 key	116	0x74
VK_F6	F6 key	117	0x75
VK_F7	F7 key	118	0x76
VK_F8	F8 key	119	0x77
VK_F9	F9 key	120	0x78
VK_F10	F10 key	121	0x79
VK_F11	F11 key	122	0x7A
VK_F12	F12 key	123	0x7B
VK_F13	F13 key	124	0x7C
VK_F14	F14 key	125	0x7D

VK_F15	F15 key	126	0x7E
VK_F16	F16 key	127	0x7F
VK_F17	F17 key	128	0x80
VK_F18	F18 key	129	0x81
VK_F19	F19 key	130	0x82
VK_F20	F20 key	131	0x83
VK_F21	F21 key	132	0x84
VK_F22	F22 key	133	0x85
VK_F23	F23 key	134	0x86
VK_F24	F24 key	135	0x87
-	Unassigned	136-143	0x88-8F
VK_NUMLOCK	NUM LOCK key	144	0x90
VK_SCROLL	SCROLL LOCK key	145	0x91
VK_OEM_FJ_JISHO	OEM Jisho	146	0x92
VK_OEM_FJ_MASSHOU	OEM Mashu	147	0x93
VK_OEM_FJ_TOUROKU	OEM Touroku	148	0x94
VK_OEM_FJ_LOYA	OEM Loya	149	0x95
VK_OEM_FJ_ROYA	OEM Roya	150	0x96
-	Unassigned	151-159	0x97-9F
VK_LSHIFT	Left SHIFT key	160	0xA0
VK_RSHIFT	Right SHIFT key	161	0xA1
VK_LCONTROL	Left CONTROL key	162	0xA2
VK_RCONTROL	Right CONTROL key	163	0xA3
VK_LMENU	Left MENU key	164	0xA4
VK_RMENU	Right MENU key	165	0xA5
VK_BROWSER_BACK	Browser Back key	166	0xA6
VK_BROWSER_FORWARD	Browser Forward key	167	0xA7
VK_BROWSER_REFRESH	Browser Refresh key	168	0xA8
VK_BROWSER_STOP	Browser Stop key	169	0xA9
VK_BROWSER_SEARCH	Browser Search key	170	0xAA
VK_BROWSER_FAVORITES	Browser Favorites key	171	0xAB
VK_BROWSER_HOME	Browser Start and Home key	172	0xAC
VK_VOLUME_MUTE	Volume Mute key	173	0xAD
VK_VOLUME_DOWN	Volume Down key	174	0xAE
VK_VOLUME_UP	Volume Up key	175	0xAF
VK_MEDIA_NEXT_TRACK	Next Track key	176	0xB0
VK_MEDIA_PREV_TRACK	Previous Track key	177	0xB1
VK_MEDIA_STOP	Stop Media key	178	0xB2
VK_MEDIA_PLAY_PAUSE	Play/Pause Media key	179	0xB3
VK_LAUNCH_MAIL	Start Mail key	180	0xB4
VK_LAUNCH_MEDIA_SELECT	Select Media key	181	0xB5

VK_LAUNCH_APP1	Start Application 1 key	182	0xB6
VK_LAUNCH_APP2	Start Application 2 key	183	0xB7
-	Reserved	184-185	0xB8-B9
VK_OEM_1	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the ';;' key	186	0xBA
VK_OEM_PLUS	For any country/region, the '+' key	187	0xBB
VK_OEM_COMMA	For any country/region, the ',' key	188	0xBC
VK_OEM_MINUS	For any country/region, the '-' key	189	0xBD
VK_OEM_PERIOD	For any country/region, the '.' key	190	0xBE
VK_OEM_2	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the '/' key	191	0xBF
VK_OEM_3	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the '~' key	192	0xC0
VK_ABNT_C1	Abnt C1	193	0xC1
VK_ABNT_C2	Abnt C2	194	0xC2
-	Reserved	195-215	0xC3-D7
-	Unassigned	216-218	0xD8-DA

VK_OEM_4	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the '[' key	219	0xDB
VK_OEM_5	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the '\ ' key	220	0xDC
VK_OEM_6	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the '}]' key	221	0xDD
VK_OEM_7	Used for miscellaneous characters; it can vary by keyboard. For the US standard keyboard, the 'single-quote/double-quote' key	222	0xDE
VK_OEM_8	Used for miscellaneous characters; it can vary by keyboard. Usually the '\$!' key.	223	0xDF
-	Reserved	224	0xE0
VK_OEM_AX	Ax	225	0xE1
VK_OEM_102	Either the angle bracket key or the backslash key on the RT 102-key keyboard. '< >'	226	0xE2
VK_ICO_HELP	IcoHlp	227	0xE3

VK_ICO_00	Ico00, VK_ICO_00 virtual code produces '00' (two zeros) when pressed. Windows does not allow mapping of arbitrary Unicode codepoints to this VK code.	228	0xE4
VK_PROCESSKEY	IME PROCESS key	229	0xE5
VK_ICO_CLEAR	IcoClr key	230	0xE6
VK_PACKET	Used to pass Unicode characters as if they were keystrokes. The VK_PACKET key is the low word of a 32-bit Virtual Key value used for non-keyboard input methods. For more information, see Remark in KEYBDINPUT, SendInput, WM_KEYDOWN, and WM_KEYUP	231	0xE7
-	Unassigned	232	0xE8
VK_OEM_RESET	OEM Reset key	233	0xE9
VK_OEM_JUMP	OEM Jump key	234	0xEA
VK_OEM_PA1	OemPa1	235	0xEB
VK_OEM_PA2	OemPa2	236	0xEC
VK_OEM_PA3	OemPa3	237	0xED
VK_OEM_WSCTRL	OEM WsCtrl	238	0xEE
VK_OEM_CUSEL	OEM Cu Sel	239	0xEF
VK_OEM_ATTN	OEM Attn	240	0xF0
VK_OEM_FINISH	OEM Finish	241	0xF1
VK_OEM_COPY	OEM copy	242	0xF2
VK_OEM_AUTO	OEM Auto	243	0xF3
VK_OEM_ENLW	OEM Enlw	244	0xF4
VK_OEM_BACKTAB	OEM Back Tab	245	0xF5
VK_ATTN	Attn key	246	0xF6
VK_CRSEL	CrSel key	247	0xF7
VK_EXSEL	ExSel key	248	0xF8
VK_EREOF	Erase EOF key	249	0xF9
VK_PLAY	Play key	250	0xFA
VK_ZOOM	Zoom key	251	0xFB
VK_NONAME	NoName	252	0xFC
VK_PA1	PA1 key	253	0xFD
VK_OEM_CLEAR	Clear key	254	0xFE

GTASA_CRC32 hashing algorithm

This type of hashing is used to convert model and texture names into a small datum (4-byte integer) in memory. They are identified subsequently by comparison with existing hashes. Character keypresses are also hashed and compared to hashes of cheat strings.

C++ code

```
const unsigned long GTASA_CRC32_table[256] = {
    0x00000000, 0x77073096, 0xEE0E612C, 0x990951BA,
    0x076DC419, 0x706AF48F, 0xE963A535, 0x9E6495A3,
    0x0EDB8832, 0x79DCB8A4, 0xE0D5E91E, 0x97D2D988,
    0x09B64C2B, 0x7EB17CBD, 0xE7B82D07, 0x90BF1D91,
    0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE,
    0x1ADAD47D, 0x6DDDE4EB, 0xF4D4B551, 0x83D385C7,
    0x136C9856, 0x646BA8C0, 0xFD62F97A, 0x8A65C9EC,
    0x14015C4F, 0x63066CD9, 0xFA0F3D63, 0x8D080DF5,
    0x3B6E20C8, 0x4C69105E, 0xD56041E4, 0xA2677172,
    0x3C03E4D1, 0x4B04D447, 0xD20D85FD, 0xA50AB56B,
    0x35B5A8FA, 0x42B2986C, 0xDBBBC9D6, 0xACBCF940,
    0x32D86CE3, 0x45DF5C75, 0xDCD60DCF, 0xABD13D59,
    0x26D930AC, 0x51DE003A, 0xC8D75180, 0xBFDD06116,
    0x21B4F4B5, 0x56B3C423, 0xCFBA9599, 0xB8BDA50F,
    0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924,
    0x2F6F7C87, 0x58684C11, 0xC1611DAB, 0xB6662D3D,
    0x76DC4190, 0x01DB7106, 0x98D220BC, 0xEFD5102A,
    0x71B18589, 0x06B6B51F, 0x9FBBE4A5, 0xE8B8D433,
    0x7807C9A2, 0x0F00F934, 0x9609A88E, 0xE10E9818,
    0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01,
    0x6B6B51F4, 0x1C6C6162, 0x856530D8, 0xF262004E,
    0x6C0695ED, 0x1B01A57B, 0x8208F4C1, 0xF50FC457,
    0x65B0D9C6, 0x12B7E950, 0x8BBEB8EA, 0xFCB9887C,
    0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3, 0xFBD44C65,
    0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2,
    0x4ADFA541, 0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB,
    0x4369E96A, 0x346ED9FC, 0xAD678846, 0xDA60B8D0,
    0x44042D73, 0x33031DE5, 0xAA0A4C5F, 0xDD0D7CC9,
    0x5005713C, 0x270241AA, 0xBE0B1010, 0xC90C2086,
    0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F,
    0x5EDEF90E, 0x29D9C998, 0xB0D09822, 0xC7D7A8B4,
    0x59B33D17, 0x2EB40D81, 0xB7BD5C3B, 0xC0BA6CAD,
    0xEDB88320, 0x9ABFB3B6, 0x03B6E20C, 0x74B1D29A,
    0xEAD54739, 0x9DD277AF, 0x04DB2615, 0x73DC1683,
    0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8,
    0xE40ECF0B, 0x9309FF9D, 0x0A00AE27, 0x7D079EB1,
    0xF00F9344, 0x8708A3D2, 0x1E01F268, 0x6906C2FE,
    0xF762575D, 0x806567CB, 0x196C3671, 0x6E6B06E7,
    0xFED41B76, 0x89D32BE0, 0x10DA7A5A, 0x67DD4ACC,
    0xF9B9DF6F, 0x8EBEEFF9, 0x17B7BE43, 0x60B08ED5,
    0xD6D6A3E8, 0xA1D1937E, 0x38D8C2C4, 0x4FDDF252,
    0xD1BB67F1, 0xA6BC5767, 0x3FB506DD, 0x48B2364B,
    0xD80D2BDA, 0xAF0A1B4C, 0x36034AF6, 0x41047A60,
    0xDF60EFC3, 0xA867DF55, 0x316E8EEF, 0x4669BE79,
    0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236,
    0xCC0C7795, 0xBB0B4703, 0x220216B9, 0x5505262F,
    0xC5BA3BBE, 0xB2BD0B28, 0x2BB45A92, 0x5CB36A04,
    0xC2D7FFA7, 0xB5D0CF31, 0x2CD99E8B, 0x5BDEAE1D,
    0x9B64C2B0, 0xEC63F226, 0x756AA39C, 0x026D930A,
    0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713,
    0x95BF4A82, 0xE2B87A14, 0x7BB12BAE, 0x0CB61B38,
    0x92D28E9B, 0xE5D5BE0D, 0x7CDCEFB7, 0x0BDBDF21,
```

```

        0x86D3D2D4, 0xF1D4E242, 0x68DDB3F8, 0x1FDA836E,
        0x81BE16CD, 0xF6B9265B, 0x6FB077E1, 0x18B74777,
        0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C,
        0x8F659EFF, 0xF862AE69, 0x616BFFD3, 0x166CCF45,
        0xA00AE278, 0xD70DD2EE, 0x4E048354, 0x3903B3C2,
        0xA7672661, 0xD06016F7, 0x4969474D, 0x3E6E77DB,
        0xAED16A4A, 0xD9D65ADC, 0x40DF0B66, 0x37D83BF0,
        0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9,
        0xBDBDF21C, 0xCABAC28A, 0x53B39330, 0x24B4A3A6,
        0xBAD03605, 0xCDD70693, 0x54DE5729, 0x23D967BF,
        0xB3667A2E, 0xC4614AB8, 0x5D681B02, 0x2A6F2B94,
        0xB40BBE37, 0xC30C8EA1, 0x5A05DF1B, 0x2D02EF8D
    };

    unsigned long int __cdecl GTASA_CRC32_string (unsigned char *string)
    {
        // Example of use:
        // GTASA_CRC32_string( (unsigned char*) "LANDSTK")

        register unsigned long crc;
        crc = 0xFFFFFFFF;

        while (*string)
        {
            crc = crc >> 8 ^ GTASA_CRC32_table[*string ^ (crc & 0xFF)];
            *string++;
        }

        return crc;
    }

```

Environment

The basic principles

Units

One **point** in game is supposed to be one **meter**. That`s the way all the factors should be scaled into.

A game can place objects only on limited area called a map area and this area is surrounded by an ocean.

Map limits:

X: -3000.0 to 3000.0

Y: -3000.0 to 3000.0

Distance

While playing a game knows coordinates of player and searches for objects around to render. It is good to know game loads collisions for less than **180.0** points around player (not camera location) and loads additional **LOD** objects if distance between player and target object is higher than **300.0** map points.

Following table presents it:

Player	Distance between is higher than 180.0 and	Target COL collision	Collision isn`t working there. For example camera was attached on vehicle and this vehicle drove too far away, then it will collapse even though model is visible yet.
Player	Distance between is higher than 300.0 and	Target model	LOD (low quality) object will be loaded and shown in game

Basic map file - external files to load

The game loads two text files whose names are hardcoded – **data\default.dat** & **data\gta.dat**.

Let`s name it “Basic map file”.

These files refer to other files in game directories to load such as IMG archives with binary models, textures or collisions, COL files, item definition files, item placement files and so on. Paths used in basic map entries are relative to executable location.

default.dat is the first Basic map file loaded. It is followed by gta.dat.

It is important for game to load IMG archives before IDE using models and textures, and IDE before IPL using entries from IDE, because the game loads them in order as it is in basic map file.

Lines can be commented by the # sign at the beginning.

IMG

IMG archive contains:

- **.dff** models
- **.txd** texture archives
- **.col** collision archives
- **.ipl** binary file, map placement
- **.dat** cutscene camera movements
- **.ifp** animations for peds or objects
- **.rrr** path information for mission script
- **.scm** scripts
- **.cut** cutscene text data, properties

Example *IMG*:

```
IMG DATA\PATHS\  
CARREC.IMG
```

IDE

These entries link to item definition files which define objects which could be used in IPL.

Example *IDE*:

```
IDE DATA\  
VEHICLES.IDE
```

COLLISION

This keyword is used to define external collision files which are placed in game directories.

Number sets up what map part COL belongs to. It is town number.

Number of COL map part	Description
0	whole map, city of the city
1	first town (usually Los Santos)
2	second town (usually San Fierro)
3	third town (usually Las Venturas)

IPL and MAPZONE

These keywords are used to link to IPL-style item placement and zone files. Although MAPZONE is usually used for .zon files with zone corners it is parsed the same way as IPL.

Example *IPL*:

Example *MAPZONE*:

```
MAPZONE DATA\  
MAP.ZON
```

TEXDICTION

These entries link to external, mostly generic model files.

Example *TEXDICTION*:

```
TEXDICTION MODELS\GENERIC\  
WHEELS.TXD
```

MODELFILE

These entries link to external, mostly generic texture archives.

Example *MODELFILE*:

```
MODELFILE MODELS\GENERIC\  
WHEELS.DFF
```

HIERFILE

There is no hier file used in standard GTA San Andreas and structure is unknown.

Example *HIERFILE*:

```
HIERFILE MODELS\  
EXAMPLE.HIER
```

EXIT

This command stops any further processing of the gta.dat file.

Example *EXIT*:

```
EXIT
```

Declaring objects (IDE and dynamic object files)

Every object needs to be declared before the use. Dynamic objects which could be moved or destroyed have to be described in special files.

IDE files - item definition

They declare each item which must have unique ID, not declared before. Each item is identified by unique ID in range of **0-20000**. Firstly [IDE files](#) need to be defined in basic map files; it is described on page 43.

Trying to use already occupied ID will crash the game.

IDE files are split into sections and their format is as follows:

```
# This line is commented and skipped by the GTA SA parser
objs
16000, drvin_screen, con_drivein, 150, 4
16001, drvin_projhut, con_drivein, 100, 0
end
anim
16776 ,des_cockbody ,desn2_peckers ,countn2 ,290 ,0
16777 ,des_stmotsigbas1 ,des_southtown ,countn2 ,200 ,0
16778 ,des_ufosign ,des_ufoinn ,countn2 ,150 ,0
16779 ,ufo_light02 ,ufo_bar ,countn2 ,20 ,4
16780 ,ufo_light03 ,ufo_bar ,countn2 ,20 ,4
16781 ,cn2_ringking ,des_stownstrip2 ,countn2 ,150 ,128
16782 ,a51_radar_scan ,a51_detailstuff ,countn2 ,20 ,4
end
```

Comments are started with a **#** sign at a beginning of the line and ignored by the GTA SA parser.

CARS

This section is used to define vehicles and some of their properties.

All vehicles	car, trailer, quad, mtruck, bmx and bike. NOT boats
%d Id, %s ModelName, %s TxdName, %s Type, %s HandlingId, %s GameName, %s Anims, %s Class, %d Frequency, %x Flags, %d Comprules,	%d WheelID, %f WheelScale_Front, %f WheelScale_Rear, %d Argument15

Explanation of arguments:

Argument	Type	Description
ID	integer	Unique object ID
ModelName	string	Name of the .dff model file without extension.
TxdName	string	Name of the .txd texture dictionary without extension.
Type	string	Type of vehicle, given as

		string. Types.
HandlingId	string	Name corresponding to its handling data in the handling.cfg file.
GameName	string	Name corresponding to its GXT entry, case sensitive and must be 7 characters or less. Invalid name will not show up in the game.
Anims	string	Name of the .ifp animation file without extension. It is mainly used on bikes. null to ignore this argument and use default animation.
Class	string	Class of the vehicle - it specifies the class of driver which will be created in car if car has been spawned in random traffic. GTA SA will search for ped with that class as a driver.
Frequency	integer	Influences a frequency of the vehicle spawning randomly on the streets.
WheelID	integer	Wheel index, needs to be -1 for vehicles to use wheel model defined in the vehicle's model (wheels.dff is no longer used)
WheelScale_Front	float	Scale of front wheels and collision models for types car, trailer, quad, mtruck, bmx and bike
WheelScale_Rear	float	Scale of rear wheels and collision models for types car, trailer, quad, mtruck, bmx and bike
Argument15	integer	Unknown value. For most vehicles it is 0, or -1 regardless of Type. For the Flash, Elegy, Stratum, Jester and Uranus this value is 1. For the Remington, Blade, Slamvan, Savanna,

		Broadway and Tornado this value is 2.
--	--	--

Example:

```
cars
# car and all other vehicles except boats are defined by 15
parameters
400, landstal, landstal, car,      LANDSTAL, LANDSTK,  null,
    normal,    10,    0,    0,      -1, 0.768, 0.768,    0
# only boats are defined by 11 parameters
430, predator, predator, boat,     PREDATOR, PREDATR,  null,
    ignore,      10,    0,    0
end
```

PATH

GTA San Andreas introduces new binary format and *path* section is already unused.

path header format
%d Argument1, %d Argument2, %s Argument3

Sscanf pattern is messed up in GTA SA and good pattern from GTA VC is described below.

path data format
%d Type, %d Next, %d Argument3, %f directionX, %f directionY, %f directionZ, %f Median, %d LeftTrafficLanes, %d RightTrafficLanes, %d SpeedLimit, %d Access, %f SpawnFrequency

Explanation of arguments:

Argument	Type		Description
Type	integer	0 1 2	Null node. External node. Internal node.
Next	integer	-1 0 to 11	Do not link to any other node in this group. Link to this node number in this group.
Argument3	integer	0	Unknown argument with value 0 in all standard path entries.
directionX	float	-48000.0 to 48000.0	(X) Position of node East from environment center multiplied by 16.0
directionY	float	Above 100.0 (lowest height, player is respawne	(Y) Position of node South from environment center multiplied by 16.0

directionZ	float	d below)	(Z) Position of node upwards from environment center multiplied by 16.0
Median	float	More than 0.0	Beware real width from game map is multiplied by 16.0 here.
			For vehicles: Width of the divider between left lanes and right lanes
			For peds: Width of the line they walk down.
			For boats: Unknown purpose.
LeftTrafficLanes	integer		Number of traffic lanes to the left of this node.
RightTrafficLanes	integer		Number of traffic lanes to the right of this node.
SpeedLimit	integer	Values:	The node speed limit
		0	slow - typical polish road
		1	medium - in cities
		2	fast - on highways
Access	integer	1st bit: 1	backroad bit, cars don't drive onto the backroads (turn around if necessary) and cop cars don't get spawned on them either when the player has wanted stars.
		2nd bit: 2	police roadblock bit, police can set-up roadblocks only on nodes with the 2nd bit set; this flag gets also accessed when the game sets-up traffic lights on start-up so there might be interdependencies there.
		3rd bit: 4	restricted access area. Cars with a certain flag set (at +168 into the vehicle block, 3rd bit ("4")) cannot enter the path just as normal vehicles do not enter backroads (see above). You can set the flag to true for a specific vehicle with the 0428 opcode
SpawnFrequency	float	More than 0.0 to 1.0	Spawn frequency, 0.0 means never spawn, 1.0 means always spawn when

			the node has been chosen for a spawn
--	--	--	--------------------------------------

General appearance

There are some things which make a game look better or worse. Although every model and texture how the game looks, there are still some some files controlling a world independly of where is player is player. It concerns to sky or particles which appear almost everywhere.

EnbSeries

Introduction

ENBSeries is graphical plugin made by Russian modder **BorisVorontsov**.

It adds car reflections, screen space ambient occlusions and indirect lightning (SSAO, SSIL), bloom, motion blur, shadows, water to GTA San Andreas.

Too see how much it changes GTA SA appearance compare a pictures below:

Without ENBSeries:



With EnbSeries



Configuration of enbseries.ini is stored in enbseries.ini file and can be edited in any of text editors.

PROXY

EnableProxyLibrary [boolean] - when activated, load 3rd party library (other modification) by the ENBSeries at game start, path to it written in parameter ProxyLibrary. This allow to solve problem with multiple d3d9.dll files that can't be used at once without special loaders, but if their internal code was made not correct to handle loading not directly by game, this will result in game crash or just not workable. If you can't load some useful library with ENBSeries, there is an other way, to replace original d3d9.dll in Windows System32 folder with ENBSeries d3d9.dll, but it's for advanced users, who knows how to bypass file recovering system, this way 3rd party d3d9.dll will be in game folder and will load ENBSeries like original library and ENBSeries must to load that original d3d9.dll as proxy from some other path user defined, in this case InitProxyFunctions must to be set 1.

InitProxyFunctions [boolean] - connect to functions of 3rd party library, if it's true d3d9.dll library that initialize some 3d objects inside or modify them. Some modifications don't need this to be enabled, they activating when loading, but if proxy library not works, try to set this parameter to 1.

ProxyLibrary [boolean] - file name of 3rd party library. May be full path, but it must not contain special symbols, unicode characters (japanese, chinese) and limited by length. Only one library allowed currently.

GLOBAL

It configures general settings of EnbSeries.

Explanation of parameters:

UseEffect [boolean] - if set to 1, ENBSeries will be automatically enabled when a game is opened or resumed after the minimization.

AlternativeDepth [boolean] - when this is enabled, information about the image depth for some of the effects are rendered through the mist, which can improve performance in certain situations, but not all video cards draw fog with full precision. Only DX10 and DX11 cards can do it - GF 8xxx, 9xxx, GTX and Radeon HD 2000-5970. If you see large lines on OBJECTS, disable this parameter.

AllowAntialias [boolean] - enables antialiasing setting from game also on the effects of ENB. If antialiasing is enabled or disabled in the panel display drivers, this parameter will not work, turn off the antialiasing setting.

BugFixMode [integer] [0...5] - every value fixes the unsupported option or a bug in driver or hardware. Values from 0 to 5 are HDR texture formats: 0 (R32G32F)-high quality and average yield, 1 (R32F)-high quality and high efficiency, 2 (A32R32G32B32F)-high quality and low productivity, 3 (R16F)-low quality and very high performance, 4 (R16G16F)-low quality and high efficiency, 5 (A16R16G16B16F)-low quality and middle performance.

SkipShaderOptimization [boolean] - disables optimization when compiling shader, may help to eliminate errors. Try to enable this parameter if you see any artifacts.

QuadVertexBuffer [boolean] - disables optimization when compiling shader, may help to eliminate errors. Try to enable this parameter if you see any artifacts.

EnableShaders_3_0 [boolean] - some effects look better on when rendered on shaders 3.0, it is recommended to enable this parameter if your graphics card supports shaders 3.0

AdditionalConfigFile [string] - if following file exists, then EnbSeries will automatically load parameter from that file instead of default enbseries.ini. It is useful when you are making configurations for other users while your enbseries.ini still exists and could be used when additional config file is removed or renamed.

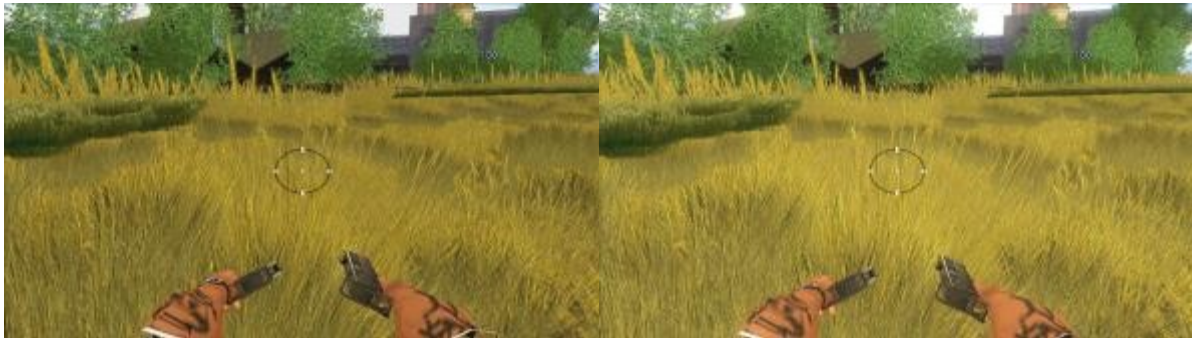
EFFECT

This section enables or disables effects. More informations below:

EnableBloom [boolean] - enables or disables bloom effect. Example on images.

Disabled bloom

Enabled bloom



Colors of rye and plant are brighter on second picture.

EnableOcclusion [boolean] - enables screen space ambient occlusions (ssao) and screen space indirect lighting (ssil), this makes shadows and lighting between nearest objects. This effect is used by few modern games, mostly on next-gen game consoles, so it's very slow. Performance directly depends from display resolution, number of pixels drawn on screen, so to run this at HD resolutions you need modern videocard from high price category. It requires support of shaders 3_0, but even if you have it, it's not guarantee fast performance. If too slow for you, reduce display resolution and quality of this effect, even disable it. Hardware compatibility is limited, different drivers and videocards have problems and limited features that result in artifacts. For example antialiasing (multisampling) for HDR textures supported only by DirectX10 compatible videocards and enabled antialiasing in game or in video drivers panel will produce strange artifacts.

EnableReflection [boolean] - reflection of vehicles. Developed for GTA San Andreasm GTA 3, GTA Vice City, but in some games also works, of course not for cars. Performance highly depends from multiple quality setting of this effect and number of objects drawn. More reflective objects on screen, slower speed. Real time 3d rendering works faster when number of objects is low, but their vertex number is high, for example 10 objects with 10 millions of vertices every will draw about the same speed as 3000 objects with 3 vertices each, for reflection on shaders may be used multiple drawing of the same object. In this case performance highly depends from CPU and system memory speed. Per pixel lighting in newest versions of ENBSeries may be faster for some videocards.

See more info for [REFLECTION](#).

EnableMotionBlur [boolean] - blurring image in fast motion of camera. On modern videocard not too much decrease speed, but costly. Current version of this effect has many problems, not working in most games, wrong, affect HUD, later will it be changed. On some videocards does not work correctly, if you have some problems, try to disable.

EnableWater [boolean] - enables water effects. Depending from it's presets may affect speed very much, by itself it's fast, but for deepness factor it needs scene depth information. Good way to compencate losts for depth is to use scene depth for other effects in parallel (shadows, ssao, dof). Modified water textures or water material (object) setting may produce unpredictable results.

DepthBias [int][0..1000] - shifts objects when drawing scene depth, offset relative to camera viewpoint. For some videocards and drivers, combination of ENB Series presets, may be required to remove flickering and hiding artefacts of ambient occlusions. Try to set 100 if something wrong happens. This parameter not affect speed.

EnableDepthOfField [boolean] - enables eye focusing effect. This effect affects performance a lot, because it needs scene depth information. When used together with other effects that also require scene depth, performance is not too much affected.

See more informations for

ENGINE (to do)

INPUT

KeyUseEffect [int][1..255] - decimal key number for activation/deactivation of modification, by default F12.

KeyBloom [int][1..255] - decimal key number for bloom activation/deactivation, by default F9.

KeyReflection [int][1..255] - decimal key number for reflection activation/deactivation, by default F11.

KeyCombination [int][1..255] - decimal number of additional key for combining this key with others, by default SHIFT.

KeyScreenshot [int][1..255] - decimal key number for capturing screenshots, by default PRINTSCREEN. Images stored in the same folder where ENB Series, but not always, depending from game. Files have BMP format, 32 bit with alpha channel (not all image viewers support this).

KeyShadow [int][1..255] - decimal key number for shadow activation/deactivation, by default F8.

KeyWater [int][1..255] - decimal key number for water activation/deactivation, by default F7.

KeyShowFPS [int][1..255] - decimal key number activation/deactivation performance statistic displayed on screen, by default * (multiply).

See table of

- [illegible]

[illegible]

A game sequence of working.

REFLECTION

Short note: reflection applies only to vehicles and it is not effect that drops down the performance a lot.

ReflectionPower [int] [0...5] - Regulates power - brightness of reflections.

ReflectionPower=12



ReflectionPower=50



ChromePower [int][0.100] - temporary disabled. Level of steel vehicle parts reflection.

UseCurrentFrameReflection [boolean] - when 1 use for reflection current screen image, this reduce quality because not all objects reflected, but there is no delay between scene and reflections. Otherwise use previous frame image. Performance is much faster when this parameter is active. In GTA San Andreas game reflection level depending a bit from this parameter, so after changing it correct ReflectionPower also.

UseCurrentFrameReflection = 0

Previous frame is used.



UseCurrentFrameReflection = 1

Current frame is used.



ReflectionQuality [int] [0..2] - quality, 0 means maximal quality and slowest speed. At 0 quality reflections not very sharp because they use mipmapping. Currently this parameter not affect speed too much, but in future the difference will be huge.

ReflectionQuality = 0



ReflectionQuality = 1



ReflectionSourceSpecular [int][0..100] - percent of using "specular" material color as reflection factor ("glosiness" in exporter). Some car parts may be reflective with this setting, but works good for original car models, set this to 0 to eliminate some invalid reflections, but better modify car. This parameter mixing with ReflectionSourceTFactor to compute final reflection level.

ReflectionSourceTFactor [int][0..100] - percent of using "texture factor" as game environment map mix level ("reflection" in exporter). Some car parts may not be reflective with this parameter and on the contrary, tested and work perfectly on original cars. This parameter mixing with ReflectionSourceSpecular to compute final reflection level.

UseAdditiveReflection [boolean] - reflections added to screen car colors making bright, in some situations oversaturated reflections. 0 means more softly reflection that depends from car brightness, for bright cars reflection level is lower than for dark cars. Does not affect rendering speed.

Disabled additive reflection



Enabled additive reflection



ReflectionDepthBias [int][0..1000] - shifts reflection geometry relative to camera viewpoint. For some videocards may be useful to remove flickering and hiding artifacts. ATI videocards users must set this parameter to 100 or something like that, unless they use single pass reflections.

UseLowResReflection [boolean] - use small and blurred texture as reflection, looks like matte reflection, interesting effect. By rendering speed it's a bit faster, but not too much.

Disabled

Enabled



ReflectionSinglePass [boolean] - draw reflection together with car geometry. At this moment also activates per pixel lighting, which increase vertex processing performance, but decrease it for pixel processing, so it depending from the size of car on screen. Temporary some limitations have place, shaders 3_0 required and environment texture that simulate reflections not in use. If you have artifacts on cars, try to disable this.

Disabled

Enabled



When it is enabled and graphics card does not support Pixel Shaders 3.0, you will enjoy following artifacts:



UseEnvBump [boolean] - allow deformation of reflections by car texture, brighter texels means more deformed (frequently named environment bump, dudv bump). Affect rendering speed, but not too much for modern hardware.

Disabled - see the difference on glass

Enabled - see the difference



EnvBumpAmount [int][0..1000] - level of reflections deformation. Do not set too big values, if the car was not specially designed to be compatible with environmental bump effect.

EnvBumpOffset [int][0..1000] - step for generating direction and amount of bump from car texture. Bigger texture size need less value to make visible small details.

Disabled

Enabled

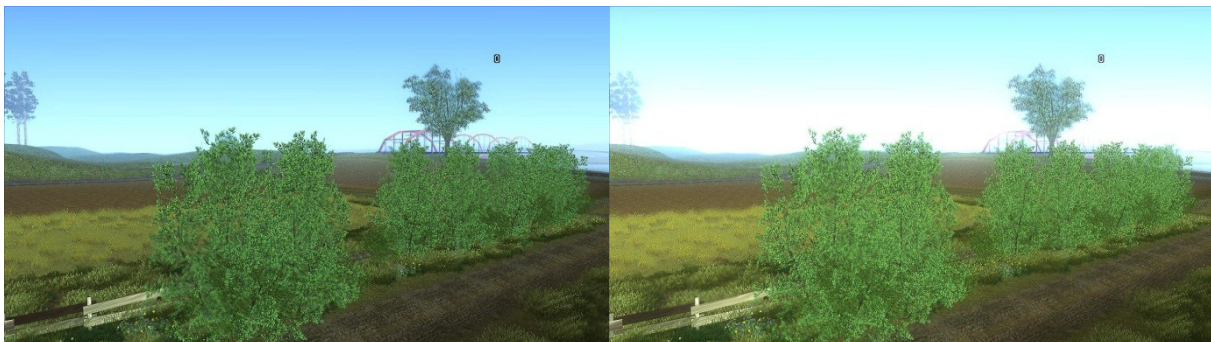


BLOOM

BloomPowerDay [int][0..100] - intensity of bloom at day time, dependent from screen brightness.

BloomPowerDay = 7

BloomPowerDay = 30

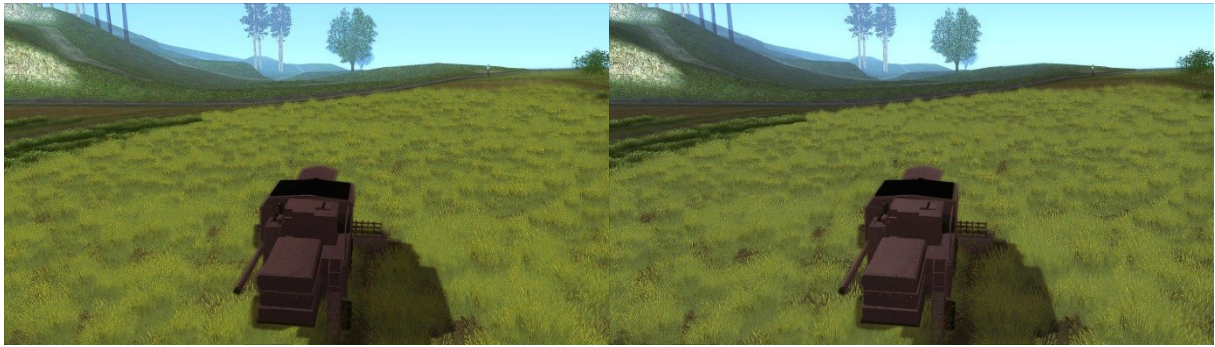


BloomFadeTime [int][0.. 100000] - time of bloom adaptation to screen brightness change, in milliseconds. Not recommended too high values, because hard to see changes in time, low values like 100 also bad idea, screen will flash frequently, very irritates.

BloomQuality [int][0..2] - bloom effect quality, 0 means maximal quality. Starting from version 0.074 speed does not much depends from this parameter, so set 0 all the time.

BloomQuality=0

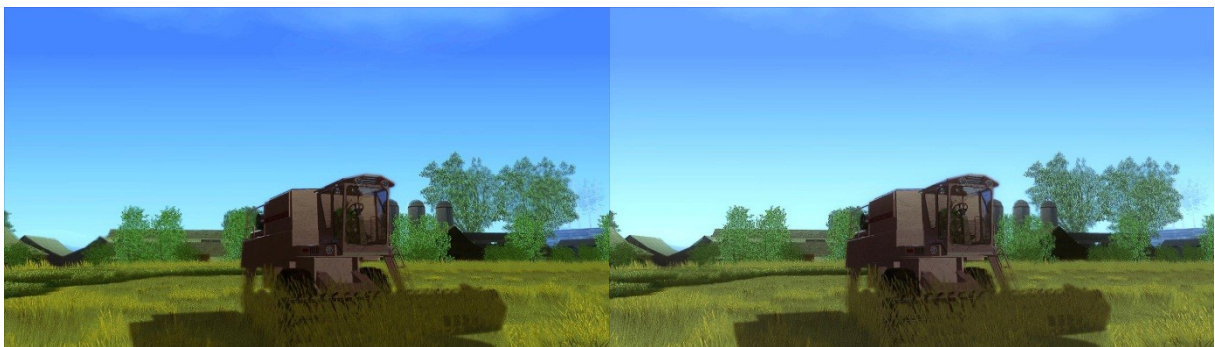
BloomQuality=2



BloomScreenLevelDay [int][0..100] - level of screen brightness in percents, that determined as day time.

BloomScreenLevelDay=30

BloomScreenLevelDay=60



BloomCurveDay [int][-10..10] - gamma correction of bloom at day time. negative values increases halftone brightness (smoggy look), positive values decrease halftones brightness (contrast, intensive image).

BloomPowerNight [int][0..100] - power of bloom at night time, dependent from screen brightness.

BloomConstantNight [int][0..100] power of bloom at night time, independent from adaptation time between screen brightness change.



BloomCurveNight [int][-10..10] - power of bloom at night time, independent from adaptation time between screen brightness change.

BloomScreenLevelNight [int][0..100] - level of screen brightness in percents, determined as night time.

BloomAdaptationScreenLevel [int][0..100] - level of screen brightness in percents, over which bloom deactivating. It's desirable that this parameter will be greater than BloomScreenLevelDay.

SSAO (to do)

UseFilter [boolean] - enable filtering of ambient occlusion texture, currently forced to be on if occlusions enabled. Ambient occlusion and indirect lighting effect use randomization for sampling textures and this produce noise. Filtering is expensive algorithm, it depends from display resolution (any filtering depends from resolution actually), so this is the one reason of slow performance, try to change it quality.

COLORCORRECTION (to do)

WATER

UseWaterDeep [boolean] - use smooth transition between different water deep levels, low deep is transparent water, deep water has color of water object. As effect itself not too high decrease speed, but it need for computation scene depth information that computes really slow, if you don't use any effects that require scene depth (ambient occlusion, depth of field, shadow quality 0 or 2), then disabling this will increase performance greatly. Otherwise, if scene depth already used, activating this parameter is almost for free (see performance tips). When this parameter deactivated, water is clear and only refraction visible (at this time).

Disabled



Enabled



WaterDeepness [int][0..1000] - factor of water semitransparency at difference deep levels. Not affect rendering speed.

WaterDeepness = 0

WaterDeepness = 200



WaterQuality - quality of water effects, 0 means maximal quality. Currently almost no difference in performance from changing this parameter, but later things will be changed.

SHADOW

ShadowFadeStart [int][0..1000] - distance, at which shadow starts to be less intensive. It must be less or equal to value of ShadowFadeEnd, if distance to shadow is less than this parameter, shadow color do not change. In GTA San Andreas shadows appears/dissapears instantly, this parameter fix this problem. In fact, shadows still have the same behavior as before, but their transparency changes. This parameter ignored if ShadowQuality set to 2, because it need scene depth information. No speed affect from this parameter.

ShadowFadeEnd [int][0..1000] - distance at which shadow disappear completely. It must be greater or equal to value of ShadowFadeStart, if distance to shadow is greater than this parameter, shadow is invisible. Other description is the same as for ShadowFadeStart.

ShadowAmountDay [int][0..100] - percent of shadows intensity in the day. Day time computes by reading brightness of rendered screen and set by ShadowScreenLevelDay parameter. If value is 0 then shadow is not visible, if value is 100, it's opaque and dark.

ShadowAmountNight [int][0..100] - percent of shadows intensity in the night. Night time computes by reading brightness of rendered screen and set by ShadowScreenLevelNight parameter. If value is 0 then shadow is not visible, if value is 100, it's opaque and dark.

ShadowScreenLevelDay [int][0..100] - level of screen brightness in percents, that determined as day time. It's easy to compute brightness in any image editing software by blurring game screenshots. For example, Adobe Photoshop in filters have Blur->Average, it produce RGB color of screen brightness, now choose one from R, G, B components that have highest value (info panel, minimal 0, maximal 255 for 8 bit per channel images), divide it by 256 and multiply by 100, result will be screen brightness in percents. If screen brightness in the game higher than this parameter, it will be day time any way, for lower value, all brightness dependent parameters will be interpolated between night and day presets.

ShadowScreenLevelNight [int][0..100] - level of screen brightness in percents, that determined as night time. Description is the same as for ShadowScreenLevelDay parameter.

ShadowQuality [int][0..2] - quality of shadows, 0 is maximal, distance from camera to shadow and vector of shadowed surface normal affects shadow blurring. At quality 1 surface normal ignored. At quality 2 everything ignored and blurring radius is constant on screen. This parameter affect performance, because quality setting 0 and 1 use scene depth information. For old videocards, integrated videochips and videocards with 64 bit videomemory bandwidth better to set this parameter to 2. Exception is only if scene depth information is used in some other effect already (ambient occlusion, water deep, depth of field).

UseShadowFilter [boolean] - enable filtering of shadows. Currently for blurring shadows i use randomizing, this produce a lot of noise that must to be filtered, but filtering eats a lot of speed, especially on old videocards, integrated videochips and videocards with 64 bit videomemory bandwidth.

FilterQuality [int][0..2] - quality of shadows filtering, 0 is maximal and slowest, filtering happens in several rendering passes. With quality 2 only one pass used. Performance varies from this parameter, be careful, also ShadowQuality affect speed of filtering, with quality of shadows 2 filtering works much faster.

ShadowBlurRange[int][0..100] - radius of blurring on the screen, too big values produce artifacts. If value is very low, performance may be faster because of texture cache.

ShadowBlurRange = 0



ShadowBlurRange = 50



ENGINE (to do)

MOTIONBLUR (to do)

PERPIXELLIGHTING (to do)

DEPTHOFFIELD

Depth of field (DOF) is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image. A larger DOF gives sharper graphics. By contrast a smaller DOF is less GPU intensive and provides more contrast between CJ (together with the vehicle, if he is driving one) and the surroundings.

DOFQuality [int][0..2] - quality of depth of field effect. 0 means maximal quality and slow performance. Higher quality makes less noisy look. This parameter affects gaming speed and directly depends from display resolution, see performance tips section.

Look on leaves:

DOFQuality = 0



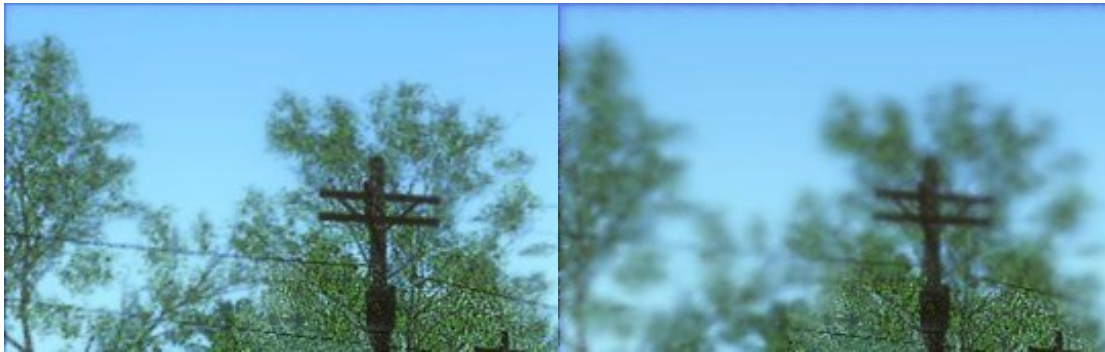
DOFQuality =2



DOFNumberOfPasses [int][1..5] - effect drawing several times, this parameter allows to configure how many times. It makes more or less visible blurriness on screen. Watch out for performance, each pass computes effect.

DOFNumberOfPasses = 1

DOFNumberOfPasses = 5



DOFFocusRange [int][0..1000] - focusing level as distance factor, less value means smaller area where objects still unblurred.



DOFBlurinessRange [int][0..10] - relative to screen radius of blurring.



Vehicles

Definition

A vehicle is a device that is designed or used to transport people or cargo. Most often vehicles are manufactured, such as bicycles, cars, motorcycles, trains, ships, boats, and aircraft.[2]

Vehicles that do not travel on land often are called craft, such as watercraft, sailcraft, aircraft, hovercraft, and spacecraft.

Land vehicles are classified broadly by what is used to apply steering and drive forces against the ground: wheeled, tracked, railed, or skied. ISO 3833- 1977 is the standard, also internationally used in legislation, for road vehicles types, terms and definitions.

Vehicles in GTA San Andreas

Types

car (0) , mtruck(1), quad(2),f_heli(3), heli(3), plane(4), boat(5), train(6), f_plane(8), bike(9), bmx(10), trailer(11)

Type and ID in memory	Description
<i>car (0)</i>	4 wheels, used for standard cars, trucks, buses, tractor. car type requires one general handling line.
<i>mtruck (1)</i>	Monster track behavior – wheels have damper and can crush the car they collide with. Wheels are scaled well for model. If model of monster truck will be used on non-mtruck vehicles, then wheels will be expanded.
<i>quad (2)</i>	Quad physics – 4 wheels bike with engine. Requires additional bike handling.
<i>f_heli (3)</i>	Unused type in standard game. The name might refer to a fighter, although setting this type to arbitrary helicopter doesn't enable shooting.
<i>heli (3)</i>	Helicopter -propeller spins around and vehicle

	can resist in the air without flying ahead unlike planes.
<i>plane (4)</i>	Planes must fly ahead in order to resist in the air, otherwise it will go down and crash on the ground.
<i>boat (5)</i>	A boat is a watercraft of any size designed to float or plane, to provide passage across water. Boats are declared by only 11 arguments inside <i>cars</i> section of IDE files.
<i>train (6)</i>	<p>This type is used for both trains and trams.</p> <p>A train is a connected series of vehicles for rail transport that move along a track (permanent way) to transport cargo or passengers from one place to another.</p> <p>A tram (also known as a tramcar, streetcar, trolley car) is a passenger rail vehicle which runs on tracks along public urban streets and also sometimes on separate rights of way. It may also run between cities and/or towns (interurbans, tram-train), and/or partially grade separated even in the cities (light rail). Trams very occasionally also carry freight.</p>
<i>f_plane (8)</i>	Unused type in standard game. The name might refer to a fighter.
<i>bike (9)</i>	A motorcycle (also called a motorbike, bike, or cycle) is a single-track, two-wheeled motor vehicle. Motorcycles vary considerably depending on the task for which they are designed, such as long distance travel,

	navigating congested urban traffic, cruising, sport and racing, or off-road conditions.
<i>bmx (10)</i>	A bicycle, also known as a bike, pushbike or cycle, is a human-powered, pedal-driven, single-track vehicle, having two wheels attached to a frame, one behind the other.
<i>trailer (11)</i>	A trailer is generally an unpowered vehicle pulled by a powered vehicle. Commonly, the term trailer refers to such vehicles used for transport of goods and materials.

Configuring vehicle

Declaring vehicle in IDE

A vehicle is one of defined objects in GTA SA, therefore it needs to be declared in IDE files.

Read informations of [CARS](#) section to find out a way to do it.

Vehicle handling - handling.cfg

The handling.cfg file is a text data file format which sets many performance and behavior values for each vehicle in GTA SA. The file can be opened and edited with any text editor.

There are also some tools to permit the user, modification of file in GUI, more readably.

File format

The handling is text file loaded line by line.

Lines can be commented by putting a ; (one semicolon) at the beginning of a line. Commented line is completely ignored and skipped by internal parser of GTA SA. Therefore it is commonly called a "single-line comment."

Explanation of sections and parameters

Standard Data

This section contains all the performance settings for land vehicles and provides the base settings for boats, bikes and flying vehicles. The below table provides the column letter, column name and a brief description for each setting available in this section.

Column n Letter	Column Name	Description
--------------------------------	--------------------	--------------------

A	vehicle identifier	Relates this data with IDE entry of vehicle.
B	fMass	Mass of the vehicle in kilograms.
C	fTurnMass	Used to calculate motion effects. Practically it changes mass acting on vehicle when it is turning (left or right) and the way how hard it is. Too high values will make vehicle able to flip around.
D	fDragMult	Changes resistance to movement, what results in maximal speed of vehicle.
F	CentreOfMass.x	Distance from the centre of the car in metres to the right for the centre of mass.
G	CentreOfMass.y	Distance from the centre of the car in metres forwards for the centre of mass.
H	CentreOfMass.z	Distance from the centre of the car in metres upwards for the centre of mass.
I	nPercentSubmerged	Percentage of the vehicle height required to be submerged for the car to float.
J	fTractionMultiplier	Cornering grip of the vehicle as a multiplier of the tyre surface friction.
K	fTractionLoss	Accelerating/braking grip of the vehicle as a multiplier of the tyre surface friction.

L	fTractionBias	Ratio of front axle grip to rear axle grip; higher value shifts grip forwards.
M	TransmissionData.nNumberOfGears	Number of gearchange animations and sound effects to use.
N	TransmissionData.fMaxVelocity	Limits the top speed.
O	TransmissionData.fEngineAcceleration	Basic rate of acceleration.
P	TransmissionData.fEngineInertia	Smooths or sharpens the acceleration curve.
Q	TransmissionData.nDriveType	Assigns Front, Rear or 4 wheel drive.
R	TransmissionData.nEngineType	Assigns Petrol, Diesel or Electric engine characteristics.
S	fBrakeDeceleration	Overall decelerative force.
T	fBrakeBias	Ratio of braking force of front compared to rear; higher values move bias forward.
U	bABS	ABS Brakes "0" = no and "1" = yes
V	fSteeringLock	Maximum angle of steering in degrees.
a	fSuspensionForceLevel	

b	fSuspensionDampingLevel	
c	fSuspensionHighSpdComDamp	Stiffens the dampening strength as speed increases.
d	suspension upper limit	Explained in my suspension tutorial.
e	suspension lower limit	Explained in my suspension tutorial.
f	suspension bias between front and rear	Ratio of suspension force to apply at the front compared to the rear.
g	suspension anti-dive multiplier	Changes the amount of body pitching under braking and acceleration.
aa	fSeatOffsetDistance	Distance from door position to seat position.
ab	fCollisionDamageMultiplier	Amount of engine damage vehicle gets from collisions. Higher value means more damage.
ac	nMonetaryValue	Used to calculate the Value of property damaged statistic.
af	modelFlags	Special animations features of the which can be enabled or disabled.
ag	handlingFlags	Special performance features.
ah	front lights	Type of head lights of the vehicle.

ai	rear lights	Same as above but for the tail lights.
aj	Vehicle anim group	Refers to an Animation ID number.

Sound properties - VehicleAudioSettings.cfg

Almost every vehicle should have sound; otherwise we would have a silent streets. In order to configure sound properties of vehicles we could use **Vehicle Audio Loader** created by *fastman92*, author of the book.

vehicleAudioSettings.cfg is text file, being processed line by line. Empty lines or lines commented by a ; sign at the beggining are ignored and skipped by the function readVehiclesAudioSettings which parses a file.

Parameters are delimited by white characters such as space or tabulator. Types of values are reserved and [float](#) 1.0 shouldn't be modified to an integer 1

Model names work for any vehicle defined inside of IDE file.

The table presents the meaning of values in line according to their position.

Line parameters

Column Letter	Column Name	Type	Description
A	modelName	string	A model name of vehicle which is defined inside of a IDE file. No matter if is standard or added vehicle.
B	VehicleType	__int16	Vehicle type, see table below.
C	EngineOnSound	__int16	Sound bank ID of accelerative, inhibitory engine.
D	EngineOffSound	__int16	Sound bank ID of stopped vehicle with enabled engine.

E	field_4	__int16	Unknown __int16. Used values by R*: 0, 1, 2
G	field_6	float	Unknown float.
H	HornTon	signed char	Not tested by the creator of plugin
I	HornHigh	float	Not tested by the creator of plugin
J	DoorSound	signed char	Not tested by the creator of plugin
K	RadioNum	char	<p>ID of default radio station being enabled when player enter a vehicle. Index of station is starting from 1.</p> <p>Click here to see the List of radio stations.</p> <p>If RadioType != 0 (civilian radio), RadioNum should equal to 13 (radio off)</p>
L	RadioType	signed char	Type of radio (-1 = no radio, 0 = civilian, 1 = special, 2 = may not exist, 3 = emergency)
M	field_14	signed char	Unknown.
N	field_16	float	Unknown float

RadioNum

This parameter determines a default radio automatically turned on when player enters a vehicle. Index of radio station is starting from **1**.

Click here to see the [List of radio stations](#).

Example

landstal	0	99	98
----------	---	----	----

0	0.779999971389771.0	7
1.0	2	8
0	0.0	

Binary file specifications

IMG archive (.img)

IMG archives contain the game files (map items), usually:

1. DFF models
2. TXD texture archives
3. COL collision sets
4. IPL binary map files
5. IFP animations for peds or objects
6. CUT cutscene text data
7. DAT cutscene camera movements

There are hardcoded IMG archives to be loaded by GTA San Andreas and additional defined in basic map files.

List of hardcoded IMG files can be found in [hardcoded img files](#) table on page **20** and description on how to define additional IMG archives in basic map file to be loaded on page **43**.

IMG archives have relatively simple format and be divided into 3 parts.

Header

<i>Offset</i>	<i>Type</i>	<i>Value</i>
0x0	CHAR[4]	"VER2" always
0x4	DWORD	Number of files on list (n)

File list

Starts on offset: 0x8

Each entry description takes: 32 bytes.

Entry description:

<i>Offset</i>	<i>Type</i>	<i>Value</i>
+0x0	DWORD	File offset, in blocks (file offset / 2048)
+0x4	WORD	Size, second priority in blocks (file size / 2048)
+0x6	WORD	Size, first priority in blocks (file size / 2048)
+0x8	CHAR[24]	File name NULL (0x00) terminated.

If +6 SizeFirstPriority is NOT zero, +6 will be used as file size, else +4 SizeSecondPriority will be used.

It's been unnoticed for 7 years because all standard R* IMG archives have +6 with zero value applied to all entries.

Zero didn't change resulting value read by IMG editors when treated as DWORD.

File contents

File contents are right after a file list. They are aligned to 2048 bytes boundaries. It means position of file in IMG archive must be divisible by 2048. Therefore file in sample archive can be stored

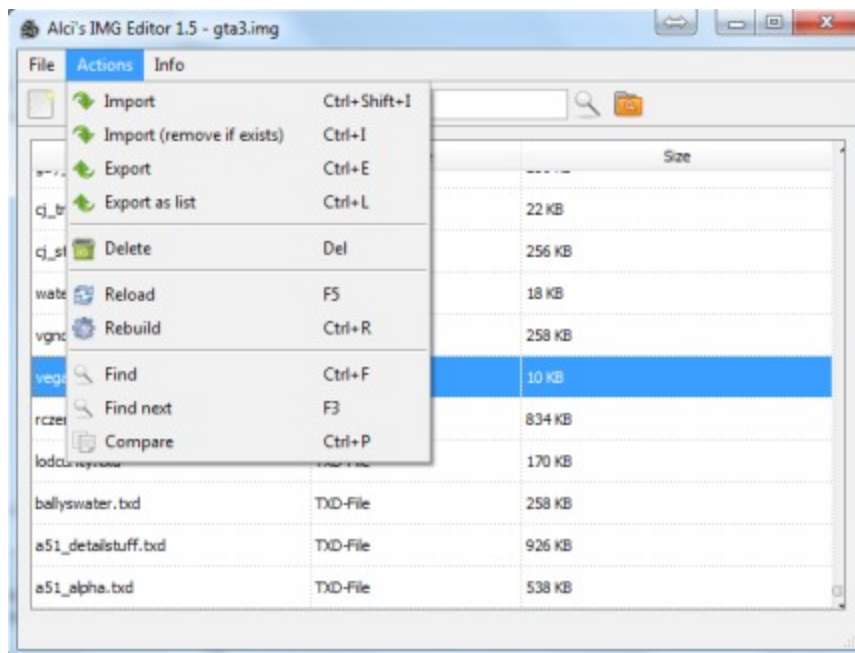
Though the file offset and size is divided by 2048 and theoretical maximum size of IMG archive is $\text{FILE_OFFSET} \times 2048$ bytes; $0xFFFFFFFF \times 2048$ bytes = 4 GB - 1 byte; $4 \text{ GB} \times 2048 = 8192$; the maximum size of IMG archive is 4 GB, because GTA San Andreas works on 4-byte integer while it multiplies by 2048 to get position of file in IMG archive and read the file.

Programs

Programs can operate on IMG archives. Good programs can remove or replace files from IMG archives without need to rebuild whole archive. They can add file, truncate IMG archive, then calculate new offsets on file list and replace them. Powerful IMG editors let the user to directly view TXD textures without need to extract them and unpack in external program. Programs are described in alphabetical order.

Alci's IMG Editor

It is IMG editor using QT GUI. Supports mostly needed operations and performs them quickly. The author is *Alci*.



Advantages:

- ✓ Creating new IMG archive
- ✓ Searching files
- ✓ Adding files
- ✓ Replacing files
- ✓ Deleting files
- ✓ Exporting files
- ✓ Rebuilding archive
- ✓ Bulk replace
- ✓ Exporting file list as text file in format (filename, newline, filename, newline, filename)
- ✓ Comparing IMG file list against previously exported file list
- ✓ No changes are applied to IMG archive until Save option is selected.

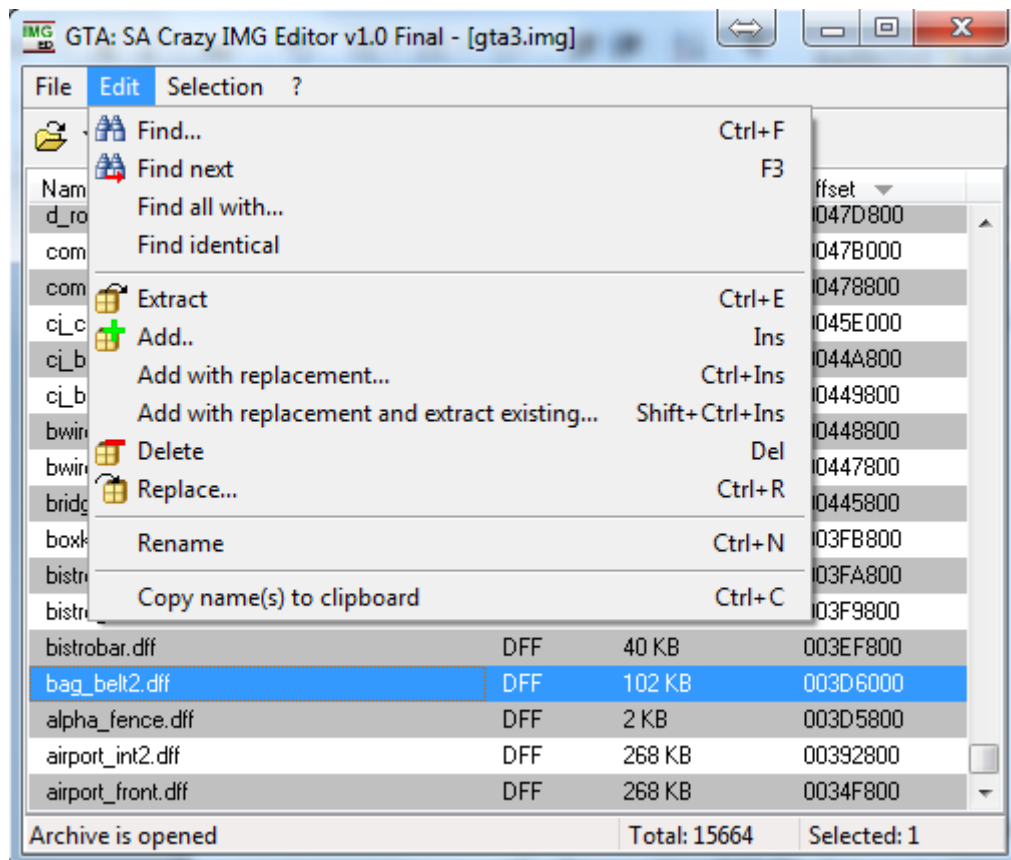
Disadvantages:

- No altering the name of files
- Need to rebuild archive.

- Doesn't support IMG archives above 2 GB.

Crazy IMG Editor

It has some unique feature like adding with replacement and extracting old files. The author is *CrazyVirus*.



Advantages:

- ✓ Searching files
- ✓ Adding files
- ✓ Replacing files
- ✓ Deleting files
- ✓ Exporting files
- ✓ Rebuilding archive
- ✓ Find identical files (by names)
- ✓ Bulk replace
- ✓ Select files with type (e.g dff files only)
- ✓ Invert selection
- ✓ Save the list of files
- ✓ Find all with part of name

Disadvantages:

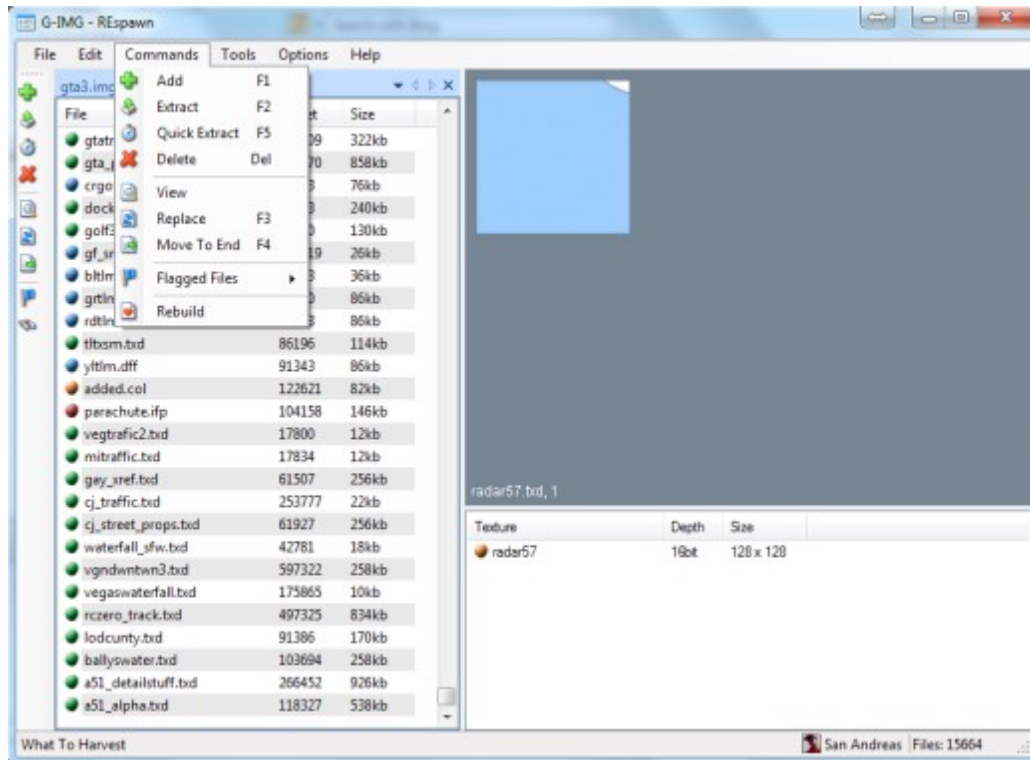
- Every change is already saved to file, can't save changes to IMG file by pressing "Save"
- Need to rebuild archive.

G-IMG

Another powerful IMG editor. The author is *RESpawn*.

Requires .NET Framework 2.0 installed. On Vista or Seven it requires to apply this fix in order to run:

`%windir%\Microsoft.NET\Framework64\v2.0.50727\ldr64.exe setwow`



Advantages:

- ✓ Creating new IMG archive
- ✓ Searching files
- ✓ Adding files
- ✓ Replacing files
- ✓ Renaming files
- ✓ Deleting files
- ✓ Exporting files
- ✓ Integrated texture viewer
- ✓ IDE Harvest
- ✓ Merging IMG archives
- ✓ Rebuilding archive
- ✓ Click double on file to open it extracted in default program
- ✓ Bulk replacement

Disadvantages:

- Every change is already saved to file, can't save changes to IMG file by pressing "Save"

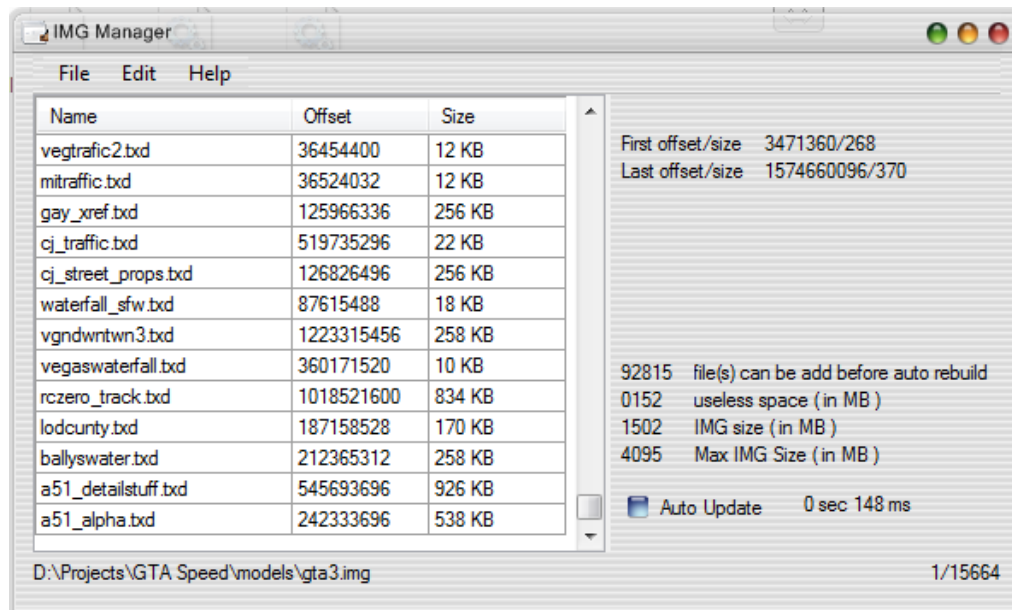
- Need to rebuild archive.
- Doesn't support IMG archives above 2 GB.

IMG Manager

The greatest feature that makes this tool innovative is it supports archives over 2 GB up to 4 GB (IMG max size). The author is *Xmen*.

Requires .NET Framework 2.0 installed. On Vista or Seven it requires to apply this fix in order to run:

```
%windir%\Microsoft.NET\Framework64\v2.0.50727\ldr64.exe setwow
```



Advantages:

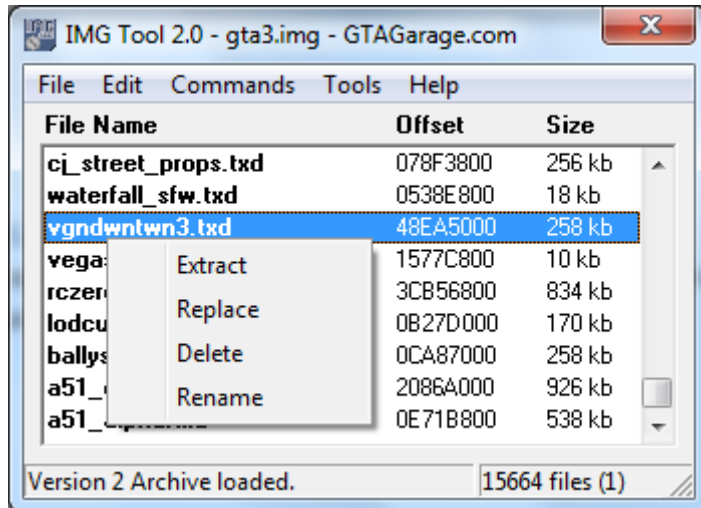
- ✓ Creating new IMG archive
- ✓ Searching files
- ✓ Adding files
- ✓ Replacing files
- ✓ Renaming files
- ✓ Deleting files
- ✓ Exporting files
- ✓ Supports IMG archives up to 4 GB size
- ✓ No need to rebuild an archive after adding file
- ✓ No need to rebuild archive
- ✓ Rebuilding archive
- ✓ Configure programs to open for COL, DFF, IFP, IPL, TXD, click double, close program and question to save updated file.
- ✓ Bulk replacement
- ✓ Drag and drop method is possible. No need to waste time on selection of folder. You can drag and drop files from explorer to Spark file list and file will be imported. Exporting by dragging and dropping is also possible for files up to 25 MB.
- ✓ Integrated binary IPL viewer

Disadvantages:

- Every change is already saved to file, can't save changes to IMG file by pressing "Save"

IMG Tool

It was the first working tool created for GTA San Andreas to open and edit IMG archives. The author is *Spooky*.



Advantages:

- ✓ Rebuilding archive
- ✓ Searching files
- ✓ Adding files
- ✓ Replacing files
- ✓ Extracting files
- ✓ Deleting files
- ✓ Renaming files
- ✓ Exporting files

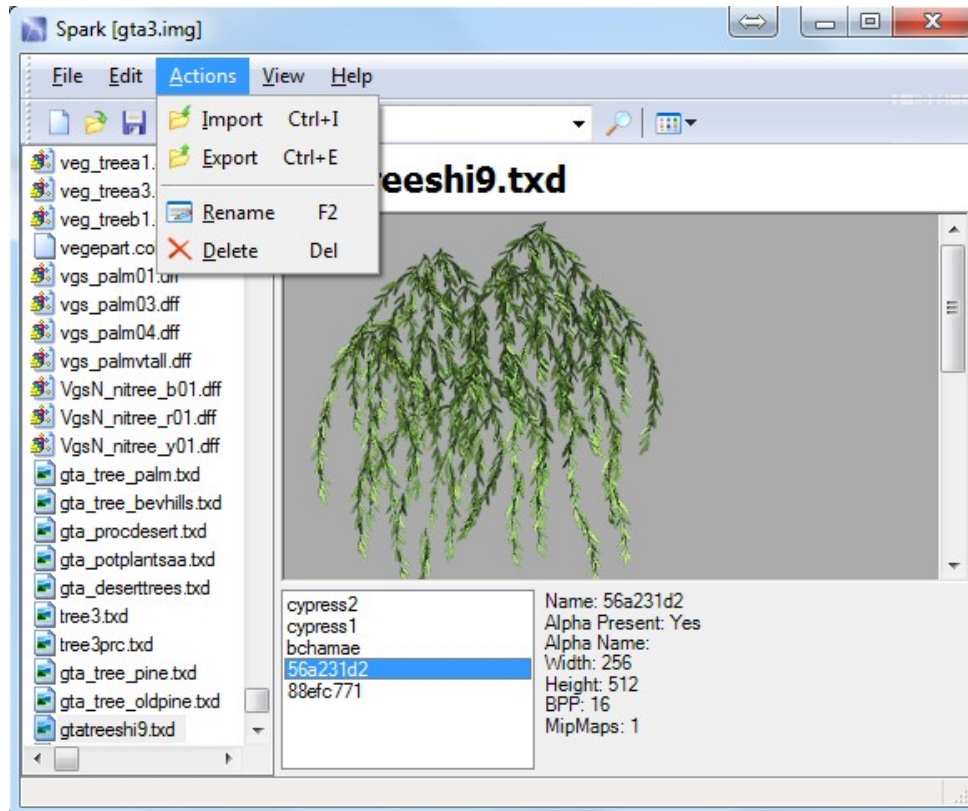
Disadvantages:

- Every change is already saved to file, can't save changes to IMG file by pressing "Save"
- Doesn't support IMG archives above 2 GB.
- Need to rebuild archive.
- No bulk replacing
- No more possibilities.

Spark IMG Editor

It is editor with a lot of features. One disadvantage is it can't rebuild archive and can't add files without need to rebuild. Program requires .NET framework 2.0

The author is Arushan (aru)



Advantages:

- ✓ Creating new empty IMG archive
- ✓ Searching files
- ✓ 4 views of file list like in explorer (Explorer, Icons, List, Details)
- ✓ Drag and drop method is possible. No need to waste time on selection of folder. You can drag and drop files from explorer to Spark file list and file will be imported. Exporting by dragging and dropping is also possible for files up to 25 MB.
- ✓ Adding files
- ✓ Replacing files
- ✓ Renaming files
- ✓ Deleting files
- ✓ Exporting files
- ✓ Bulk replacement
- ✓ Sort files by name, type, offset or size.
- ✓ Integrated texture viewer
- ✓ No changes are applied to IMG archive until Save option is selected.

Disadvantages:

- Need to rebuild archive.
- Does not support archives above 2 GB archives while 4 GB is max IMG archive size.

SCM compiled script

SCM files store missions and several other scripts.

.scm file needs special structure, while plain scripts, often with .cs extension, have an exact code only.

Compiled code - opcodes

An opcode- the smallest operation that could be executed in SCM script, including math

Compiled code - data types

Data types (in hex)	Size of data	Extra info
<u>Static numbers</u>		
01	4	Static number: (Long) Integer - 32 bits
02	2	Global 4-byte integer /float variable multiplied by 4: (Short) Integer - 16 bits
03	2	Local variable: (Short) Integer - 16 bits
04	1	Static number: (Char) Integer - 8 bits
05	2	Static number: (Short) Integer - 16 bits
06	4	Float IEEE 754 - 32 bits
<u>Number arrays</u>		
07	6	Global Array 4-byte integer /float Var: Primary Global variable ID multiplied by 4 (2 bytes): (Short) Integer - 16 bits and left 4 bytes, Compiled code - array, data next after primary variable
08	6	Local Array 4-byte integer /float Var: Primary Local variable ID (2 bytes); and left 4 bytes, Compiled code - array, data next after primary variable
<u>Short and varlength strings</u>		
09	8	(Short) string - 8 bytes
0A	2	Global 8-byte short string variable multiplied by 4: (Short) Integer - 16 bits
0B	2	Local 8-byte short string variable: (Short) Integer - 16 bits
0C	6	Global Array 8-byte short string Var: Primary Global variable ID multiplied by 4 (2 bytes): (Short) Integer - 16 bits and left 4 bytes, Compiled code - array, data next after primary variable
0D	6	Local Array 8-byte short string Var: Primary Local variable ID (2 bytes): (Short) Integer - 16 bits and left 4 bytes, Compiled code - array, data next after primary variable
0E	1 +	Varlength string:

	<i>string.length()</i>	1 st byte gives the length of exact string; therefore maximal length is 255 characters. String doesn't have to be NULL terminated, because length is known.
0F	16	(Short) string - 16 bytes
<u>Long strings</u>		
10	2	Global 16-byte long string variable multiplied by 4: (Short) Integer - 16 bits
11	2	Local 16-byte long string variable: (Short) Integer - 16 bits
12	6	Global Array 8-byte long string Var: Primary Global variable ID multiplied by 4 (2 bytes): (Short) Integer - 16 bits and left 4 bytes, Compiled code - array, data next after primary variable
13	6	Local Array 8-byte long string Var: Primary Local variable ID (2 bytes): (Short) Integer - 16 bits and left 4 bytes, Compiled code - array, data next after primary variable

Compiled code - array, data next after primary variable

Secondary Variable, index to take and add value from (2 bytes),

Size of an array (byte), amount of indexes,

Type of variable and multiplier for index to add (byte) ,

Value from secondary variable is multiplied by 4 if type is global variable.

Compiled code - array, type of multiplier and multiplier for index from secondary variable

0x00 - local variable, multiply value * 1 vars - [integer](#) arrays

0x01 - local variable, multiply value *1 vars - [float](#) arrays

0x02 - local variable, multiply value * 2 vars - [short string](#) arrays

0x03 - local variable, multiply value * 4 vars - [long string](#) arrays

0x80 - global variable, multiply value * 1 vars - [integer](#) arrays

0x81 - global variable, multiply value * 1 vars - [float](#) arrays

0x82 - global variable, multiply value * 2 vars - [short string](#) arrays

0x83 - global variable, multiply * 4 vars - [long string](#) arrays

Compiled code - array example

Example in SB:

`$5($4,42i)` - 07 14 00 10 00 2A 80

07 - initialize data type: Global Array 4-byte [integer](#) / [float](#) Var

14 00 - primary global variable multiplied by 4, $0x14/4 = \$5$

10 00 - secondary, global variable to take value from and add to primary variable, multiplied by 4, $0x14/4 = 5$

2A - size of \$5 array ($0x2A = 42$) in amount of variables

80 - according to

[Error: Reference source not found](#), 0x80 identifies secondary variable as global and multiplies * 1.

Audio

Radio

List of radio stations

Radio ID, starting from 1	Radio name
0	Undefined
1	Playback FM
2	K Rose
3	K-DST

4	Bounce FM
5	SF-UR, San Fierro Underground Radio
6	Radio Los Santos
7	Radio X
8	CSR 103.9, Contemporary Soul Radio
9	K-JAH West
10	Master Sounds 98.3
11	WCTR, West Coast Talk Radio
12	User tracks
13	Radio Off

Memory

Information

How the process in memory works.

Every running application (process) occupies memory where are stored temporarily generated data. .exe file goes to memory too. Good examples of temporarily generated data are loaded models, assigned tasks, created actor & vehicle, object handles and so forth.

Every different EXE has different memory addresses so that gta_sa.exe 1.0 addresses don't match to gta_sa.exe 1.01.

Compiled application has compiled CPU instructions called Assembler (ASM shortly). EXE file content could be divided into following parts:

- ASM code – it is executed by processor. Fully loaded to memory.
- Used data – they are in EXE and are loaded to memory, but aren't processor instructions; instead they can be processed as values (get to CPU register) during execution of ASM code.
- Resources – they are in EXE, but need to be loaded through ASM to memory. Therefore it is possible to have EXE installers of enormous size and data go to memory only by request.

Beginning of Windows executable is stored in memory on address: **0x400000**

However ASM code and values in memory are divided to allocated segments. For this part of EXE they are static so that their address is not dynamic and different during each execution of game.

EXE content is read/write protected in memory.

- Generated data – they are only in memory and are generated by ASM code based on files, calculations and so on. It is more complex than two sentences.

Read/Write protected (virtual protect) – it is important to know if address to modify is read/write protected. If it is, then it cannot be directly modified, since it needs to have a protection removed. There is available VirtualProtect function in programming languages and could be used while making trainers or ASI plugins which change the memory. CLEO opcodes reading memory can handle virtual_protect too.

Hex editors

Hex Editor with ability to open and edit process is necessary to gain deeper understanding of how memory works. Hex Editor Neo is recommended which is payable, but it is worth all spent coins.

Another hex process editor is HxD. Main disadvantage is disability to analyze data types directly in hex editor what makes analyzing values harder. In memory there is no type of data set up, but what is meaning of value on address depends on how this value will be treated by ASM code. It might be confusing as hell however there could be specified types of data in memory.

Cheat Engine is memory editor and disassembler. When address is known it can find out what address containing ASM code is reading/writing to this address.

NOP Operation:

NOP – No operation in ASM. It's single code is 0x90

Usually it is necessary to replace some ASM code with 0x90 to prevent something, for example if address is found out and it is not necessarily overwritten. Then Cheat Engine can be used to find out what writes to this address and can show how many bytes this ASM instruction have.

Value types

Name	Description	Size	Range
boolean	Stores 0/1 true or false value.	1 byte = 8 bits	0 or 1 : TRUE or FALSE
byte or char	Character or small integer.	1 byte = 8 bits	signed: -128 to 127 unsigned: 0 to 255
word, int	Short integer, word is always unsigned.	2 bytes = 16 bits	signed: -32768 to 32767 unsigned: 0 to 65535
dword, longword, int	Short signed integer. Dword is always unsigned.	4 bytes = 32 bits	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
float	Single precision floating-point format	4 bytes = 32 bits	to $(2-2-23) \times 2^{127} \approx 3.4 \times 10^{38}$
string, text	String is a finite sequence of symbols that are chosen from a set or alphabet. It is NULL (0x00) terminated.	Length is determined by NULL (0x00) which terminates string	Length is determined by NULL (0x00) which terminates string

Finding & patching memory addresses

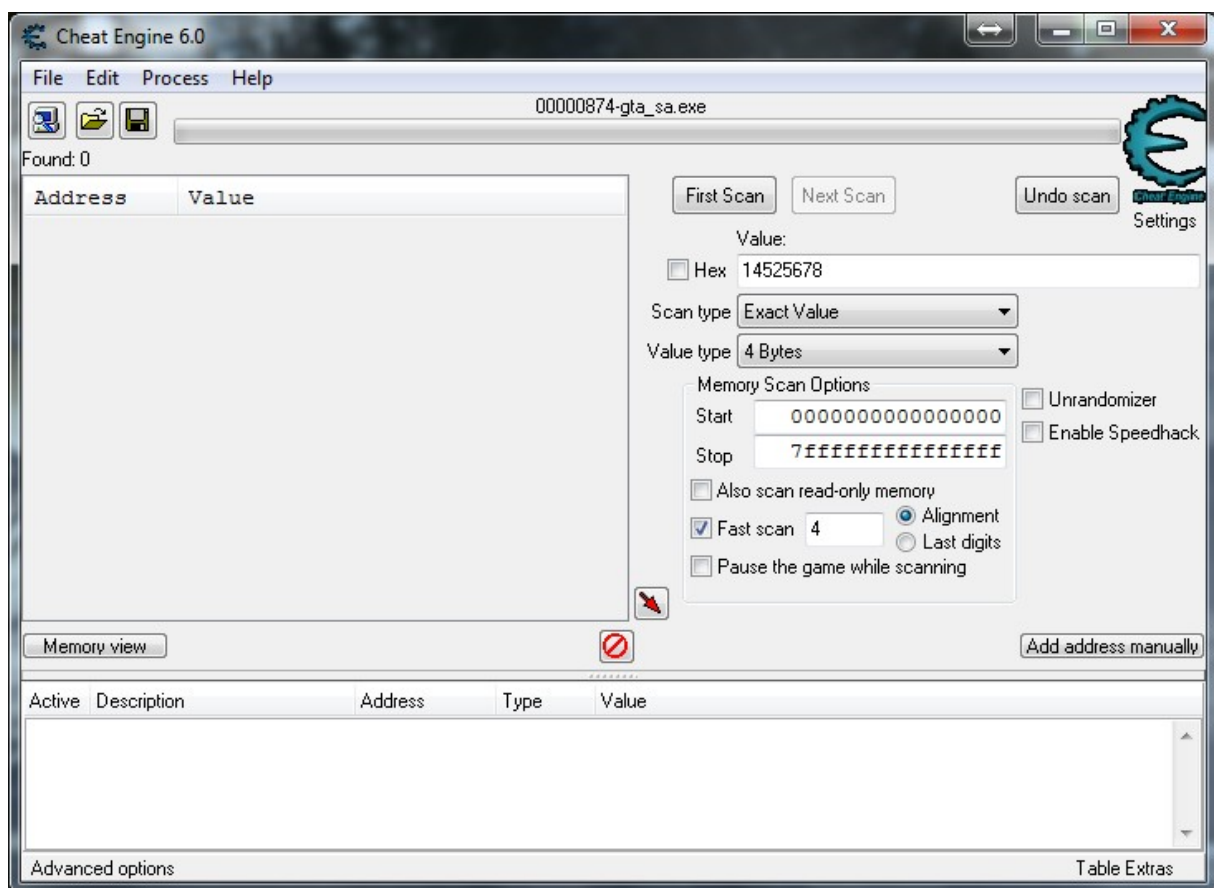
Instruction of finding addresses is designed for beginners. IDA Disassembler and solid ASM knowledge is needed for analyzing complex EXE. Cheat Engine is used to find these addresses in tutorials. Example below is performed on GTA San Andreas HOOLDRUM No-CD 1.0 US

As always the most trivial example in finding memory addresses is searching for money address. First make sure you have a big number of money, because it is not often in memory to have many the same multibyte values.

See in game what is number of your money shown on HUD. It is **14525678** in example. Memorize it and minimize game.



Open Cheat Engine, open process named gta_sa.exe. Type a money value in field



Now problem needs to be considered what is Value type. It is known that money in GTA San Andreas don't have fraction and do not look like this: 4.5; there is \$4.

Float and double drops out because they are for fractional numbers.

Text - it is just string NULL (0x00) terminated. Money won't be text value in memory as they are processed for calculations.

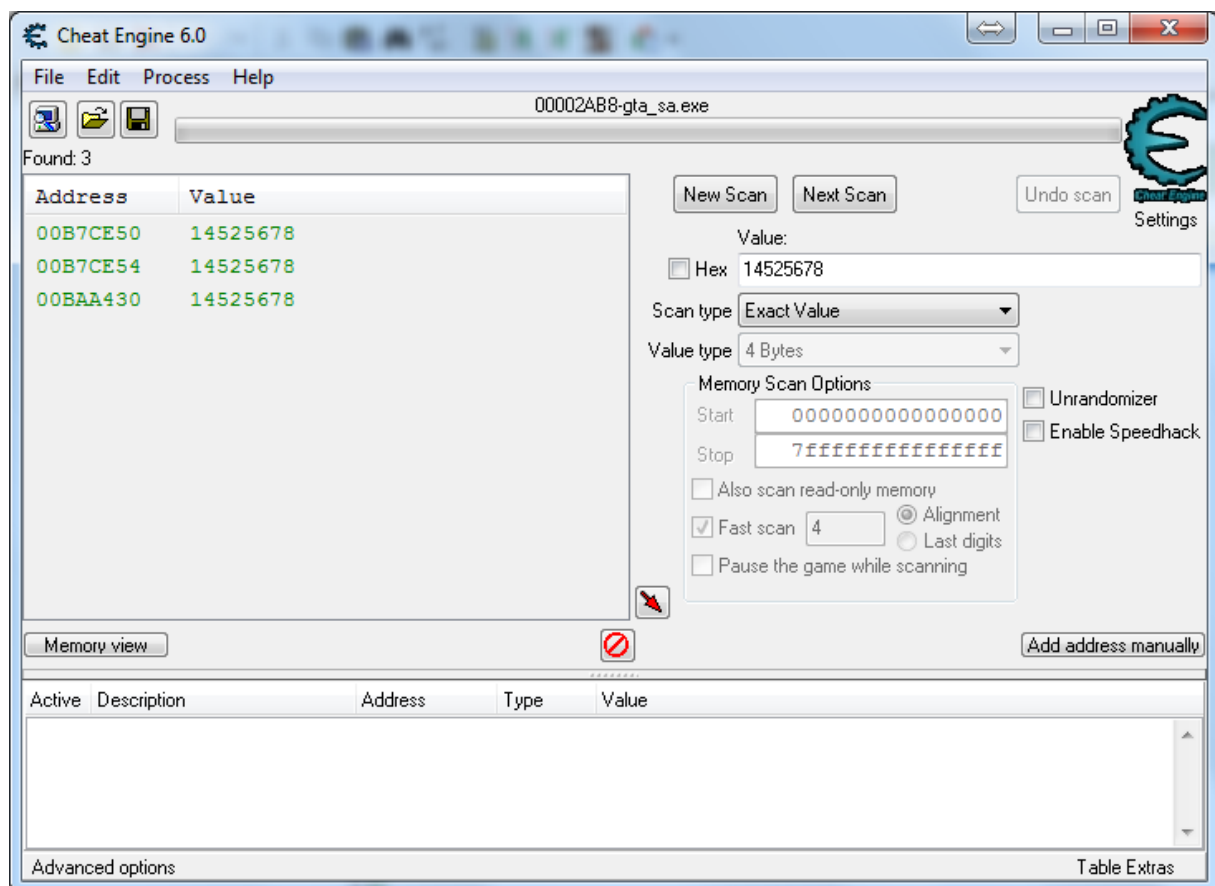
Now it is known it must be integer value. 1-byte, 2-byte or 4-byte. It tells about max number of value.

You need to recognize what is highest number of money in GTA San Andreas. It is **999 999 999**.

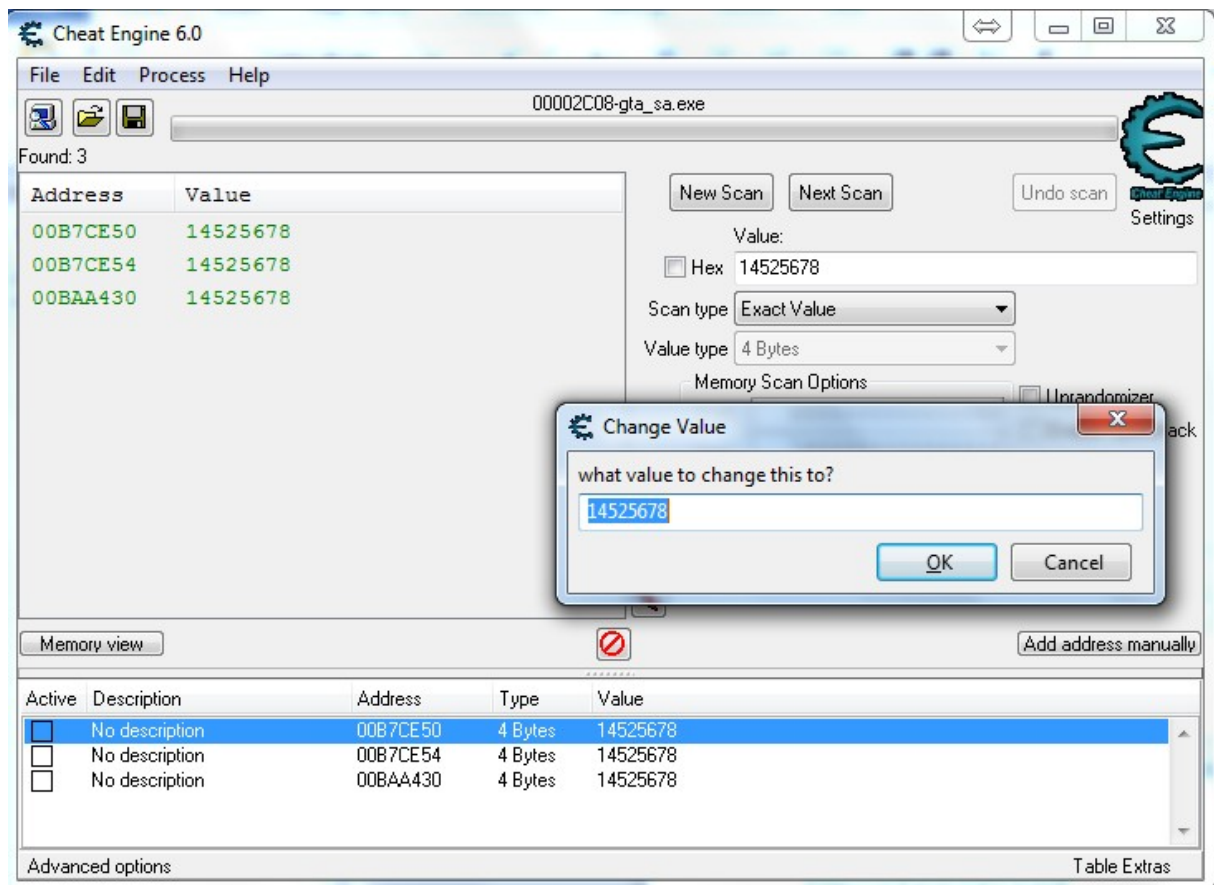
Check what type of value can store value of this size or higher by finding it on the table [Value types](#), above.

It is 4-byte integer obviously, because only 4-byte int can store numbers of minimum value **999 999 999**.

Click *First scan* in Cheat Engine.



3 values have been found. Click right red arrow to add them to the list. Now you can change one by one to check what modifies money.



Click first address value, modify and OK. Go to game. Check what happens.

Yes. That's it. Money started to count down to given value. Now what's in second address?

Modify it to e.g. 0 value. Go to game. GTA San Andreas counts down this value to value from the first address.

So that in 1.0:

First address (**0xB7CE50**) - current money

Second address (**0xB7CE54**) - money displayed on HUD, it is going to value from the first address.

Assembler short tips

xor **eax**, **eax** – reset eax, set register to 0 (zero)

and **eax**, **0FFh** – extract only first byte from eax register. Set 2nd, 3rd byte, 4th byte to zero.

Example:

	eax		
11010101	00010100 <small>sample value</small>	10111010	10101101

AND

			0xFF (255)
00000000	00000000	00000000	11111111

RESULT:

00000000	00000000	00000000	10101101
----------	----------	----------	----------

Function addresses

GTASA_CRC32_FromString (char* String)

0x0053cf00 - exe 1.0

0053D3A0 - exe 1.01

00438500 - init cheat

Memory values

Uncategorized

Identify gta_sa.exe version in memory, the first byte with different value in all of mentioned versions:

Description: Memory address to identify gta_sa.exe version

In 1.0 or 1.01: **0x400088** [dword] [virtual protect 1] [EXE Content]

Values:

1. 0xCA - GTA San Andreas v1.0 [US] HOOLDRUM No-CD Fixed EXE
2. 0x8A - GTA San Andreas v1.0 [EURO] No-CD Fixed EXE
3. 0xF9 - GTA San Andreas v1.0 [EURO] Original
4. 0xD0 - GTA: San Andreas v1.01 [EURO] No-CD/Fixed EXE

Limits

IPL Instances:

Description: The max number of active IPL instances, inst section in .ipl files. Every placed object on map is in inst section.

In Memory ([dword] [virtual protect 1] [EXE content]):

- *In 1.0:* **0x55105F**
- *In 1.01:* **0x14BE02A**

In EXE file ([dword]):

- *In 1.01:* **0xCF602A**

Streaming memory limit ([dword] [virtual protect 1] [EXE content]):

Description: Amount of memory for loading textures around camera.

When too many high-res textures are added to the game they disappear because there isn't enough memory to load. Increasing this limit solves texture loading problem.

- In 1.0: **0x8A5A80**
- In 1.01: **0x8A6D4C**

Note: This value is overwritten 3 times during the loading of game. NOP following addresses to solve a problem:

In 1.0:

- **0x5B8E64** - 10 bytes
- **0x5BCD50** - 5 bytes
- **0x5BCD78** - 5 bytes

In 1.01:

- **0x5B9644** - 10 bytes
- **0x5BD530** - 5 bytes
- **0x5BD558** - 5 bytes

Player

Money ([dword]):

Description: Base address to money in GTA San Andreas

- In 1.0: **0xB7CE50**
- In 1.01: **0xB7F4D0**

This address:

- + **0x0** = Current money value.
- + **0x4** = Displayed number of money on the game HUD

Time

Current time ([dword]):

Description: Base address to time in GTA San Andreas

- In 1.0: **0xB7014E**
- In 1.01: **0xB727CE**

This address:

- + **0x0** [byte] = Current weekday (1 through 7)
- + **0x4** [byte] = Current minute (0 to 59)
- + **0x5** [byte] = Current hour (0 to 23)
- + **0x6** [byte] = Current month day
- + **0x7** [byte] = Current month (1 through 12)
- + **0xA** [byte] = Timer related to weather and time in ms, should not be modified
- + **0xE** [dword] = Defines how many milliseconds takes one minute in game.

Weather

Weather settings:

Description: Base address to weather settings.

- In 1.0: **0xC81318**
- In 1.01: **0xC83AD4**

This address:

- + **0x0** [word] = Weather lock
- + **0x4** [word] = Upcoming weather
- + **0x8** [word] = Current weather

CActor

Actor is every spawned ped. Every spawned actor is described as object in memory. CActor contains actor properties, not model or anything else. It could be divided into two parts – main struct and pool. Main struct contains pointers + info about amount of present actors, while pool contains ped structs. Firstly pointer for main struct is necessary.

Pointer to CActor main struct ([dword]):

Description: Pointer to main struct that is described below.

- In 1.0: **0xB74490**
- In 1.01: **0xB76B10**

Main struct

- + **0** ([dword]) = Contains a pointer to the first element in the pool
 - + **4** ([dword]) = Contains a pointer to a byte map indicating which elements are in use.
 - + **8** ([dword]) = Is the maximum number of elements in the pool
 - + **12** ([dword]) = Is the current number of elements in the pool
-

Explanation of main struct:

+ **0** contains a pointer to the first element in the pool. Every spawned ped is one of objects in pool. Each ped takes 1988 (0x7C4) bytes in pool. These 0x7C4 bytes contain ped properties like health, location, tasks and so forth.

+ **4** contains a pointer to a byte map where every byte concerns state of element in pool (exists or not) and necessary information to convert ped struct into handle.

If ped exists on certain position in pool, then value on this position relative to byte map will range from **0x0** to **0x80**.

How to convert position of ped in pool/byte map to handle?

HANDLE = POSITION_IN_BYTE_MAP 256 +
VALUE_FROM_BYTE_ON_THIS_POSITION_IN_BYTE_MAP*

Let`s say byte map is on **0x9000**.

We want to convert third ped struct/position to handle that may be used in SCM opcodes.

Third ped state will be on **0x9001**, position is +1 then.

Let`s say value on **0x9002** is 0x34.

Handle = **2** * 256 + 0x34

Handle = 564

Actor struct

It describes information about every active actor in pedestrian pool. Each actor takes 1988 (0x7C4) bytes in pool.

- **+14** XYZ position structure

CVehicle - spawned vehicles

CVehicle pool is memory area where is information about all currently spawned cars. This information contains car properties, location and so on. This pool does not contain loaded model or anything.

Cheats

Cheat string ([char[30]]):

- *Description:* Buffer of last 30 typed chars on keyboard. Contains only printable characters like letters. Earlier typed chars are on next bytes.
 - *In 1.0:* **0x969110**
 - *In 1.01:* **0x96B790**
- Character set in buffer:* 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Function calls ([dwords]):

- *Description:* Function addresses when cheat gets enabled or disabled.
If cheat doesn't have a function, then value is 0x00
 - *In 1.0:* **0x8A5B58**
 - *In 1.01:* **0x8A6E40**

Hashes ([dwords])_:

- *Description:* GTASA_CRC32 hashes that are tested by initCheat function when end-user is typing on keyboard.
 - *In 1.0:* **0x8A5CC8**
 - *In 1.01:* **0x8A6FB0**

States ([dwords]):

- *Description:* Bytes with values 1/0 used to determine if toggleable cheat is currently enabled or disabled. It is used for cheats that can be both enabled and disabled.
 - *In 1.0:* **0x969130**
 - *In 1.01:* **0x96B7B0**

SCM Opcodes

List of opcodes

Camera

0003 - SET_CAM_SHAKE..... 103

Key

0A8C - write_memory..... 170

0AB0 - virtual_key_pressed..... 170

Math

0004 - set_global_int_variable..... 103

0005 - set_global_float_variable..... 104

0006 - set_local_int_variable..... 105

0007 - set_local_float_variable..... 105

0008 - add_int_value_to_global_variable..... 106

0009 - add_float_value_to_global_variable..... 106

000A - add_int_value_to_local_variable..... 107

000B - add_float_value_to_local_variable..... 108

000C - subtract_int_value_from_global_variable..... 108

000D - subtract_float_value_from_global_variable..... 109

000E - subtract_int_value_from_local_variable..... 110

000F - subtract_float_value_from_local_variable..... 110

0010 - multiply_by_int_value_in_global_variable..... 111

0011 - multiply_by_float_value_in_global_variable..... 112

0012 - multiply_by_int_value_in_local_variable..... 112

0013 - multiply_by_float_value_in_local_variable..... 113

0014 - divide_by_int_value_in_global_variable..... 114

0015 - divide_by_float_value_in_global_variable..... 114

0016 - divide_by_int_value_in_local_variable..... 115

0017 - divide_by_float_value_in_local_variable..... 116

0018 - is_int_global_variable_greater_than_value..... 116

0019 - is_int_local_variable_greater_than_value..... 117

001A - is_int_value_greater_than_global_variable..... 118

001B - is_int_value_greater_than_local_variable..... 119

001C - is_int_global_variable_greater_than_global_variable..... 120

001D - is_int_local_variable_greater_than_local_variable.....	121
001E - is_int_global_variable_greater_than_local_variable.....	122
001F - is_int_local_variable_greater_than_global_variable.....	123
0020 - is_float_global_variable_greater_than_value.....	124
0021 - is_float_local_variable_greater_than_value.....	125
0022 - is_float_value_greater_than_global_variable.....	126
0023 - is_float_value_greater_than_local_variable.....	127
0024 - is_float_global_variable_greater_than_global_variable.....	128
0025 - is_float_local_variable_greater_than_local_variable.....	129
0026 - is_float_global_variable_greater_than_local_variable.....	130
0027 - is_float_local_variable_greater_than_global_variable.....	131
0028 - is_int_global_variable_greater_or_equal_to_value.....	132
0029 - is_int_local_variable_greater_or_equal_to_value.....	133
002A - is_int_value_greater_or_equal_to_global_variable.....	134
002B - is_int_value_greater_or_equal_to_local_variable.....	135
002C - is_int_global_variable_greater_or_equal_to_global_variable.....	136
002D - is_int_local_variable_greater_or_equal_to_local_variable.....	137
002E - is_int_global_variable_greater_or_equal_to_local_variable.....	138
002F - is_int_local_variable_greater_or_equal_to_global_variable.....	139
0030 - is_float_global_variable_greater_or_equal_to_value.....	140
0031 - is_float_local_variable_greater_or_equal_to_value.....	141
0032 - is_float_value_greater_or_equal_to_global_variable.....	142
0033 - is_float_value_greater_or_equal_to_local_variable.....	143
0034 - is_float_global_variable_greater_or_equal_to_global_variable.....	144
0034 - is_float_global_variable_greater_or_equal_to_global_variable.....	145
0035 - is_float_local_variable_greater_or_equal_to_local_variable.....	146
0036 - is_float_global_variable_greater_or_equal_to_local_variable.....	147
0037 - is_float_local_variable_greater_or_equal_to_global_variable.....	148
0038 - is_int_global_variable_equal_to_value.....	149
0039 - is_int_local_variable_equal_to_value.....	150
003A - is_int_global_variable_equal_to_global_variable.....	151
003B - is_int_local_variable_equal_to_local_variable.....	152
003C - is_int_global_variable_equal_to_local_variable.....	153
0042 - is_float_global_variable_equal_to_value.....	154
0043 - is_float_local_variable_equal_to_value.....	155

0044 - is_float_global_variable_equal_to_global_variable.....	156
0045 - is_float_local_variable_equal_to_local_variable.....	157
0046 - is_float_global_variable_equal_to_local_variable.....	158
0058 - add_int_global_variable_to_global_variable.....	167
0059 - add_float_global_variable_to_global_variable.....	168
005A - add_int_local_variable_to_local_variable.....	169

NOP

0000 - NOP.....	100
0052 - NOP_floats.....	165

Player

0053 - CREATE_PLAYER.....	166
---------------------------	-----

SCM Structure opcodes

0001 - WAIT.....	101
0002 - GOTO.....	102
004D - jump_if_false.....	159
004E - TERMINATE_THIS_SCRIPT.....	160
004F - START_NEW_SCRIPT_WITH_ARGS.....	161
0050 - gosub.....	163
0051 - return.....	163

Standard opcodes

These opcodes exist in GTA San Andreas without installing anything and provide most powerful scripting in game. They provide biggest possibilities of creating missions, controlling world, including objects, peds, vehicles and so on.

0000 - NOP

Sanny Builder opcodes.txt: 0000: NOP

Sanny Builder SASCM.INI: 0000=0,NOP

Description: This opcode does absolutely nothing. 0000 is often put at the beginning of CLEO scripts to avoid jump-at-zero-offset bug

Opcode definition:

```
void NOP();
```

Parameters:

No parameters

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP // there should be at least one opcode before the first
referenced label - @label in example

:0000_nop
wait 0
0000: NOP // there is no visible effect in game

01E3: show_text_1number_styled GXT 'NUMBER' number 666 time 5000 style 1
// 666 number will appear if script works
jump @0000_nop
```

0001 - WAIT

Sanny Builder opcodes.txt: 0001: wait 0 ms

Sanny Builder SASCM.INI: 0001=1,wait %ld% ms

Sanny Builder keyword: wait (wait 0)

Description: WAIT opcode stops processing of the current thread and let the GTA SA to process other active SCM threads and execute different actions. The number specifies a minimum number of milliseconds of current thread inactivity. When a number of milliseconds is less or equal to zero, then current thread will be executed as soon as possible. GTA SA stops the current thread, executes other actions and comes back to process this thread again until it encounters WAIT command again.

Native name: WAIT

Opcode definition:

```
void wait (int time);
```

Parameters:

1) Passed: integer, time to wait expressed in milliseconds. 100 ms = 1 second

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0001_wait
0001: wait 0 ms      // Stops the thread and lets the game to perform other
actions.

01E3: show_text_1number_styled GXT 'NUMBER' number 666 time 5000 style 1
// Number will appear if script works
jump @0001_wait
```

0002 - GOTO

Sanny Builder opcodes.txt: 0002: jump @MAIN_177

Sanny Builder SASCM.INI: 0002=1,jump %1p%

Sanny Builder keyword: jump (jump @label)
 goto (goto @label)

Native name: GOTO

Description: Jumps the code to the specified offset called label. Label is negative number, offset relative to beginning of SCM file.

Opcode definition:

```
void goto (int offset);
```

Parameters:

1) Passed: integer, label - negative offset in script file. (OFFSET * -1)

Returns true or false? No.

Example in Sanny Builder

```
{ $CLE0 }
0000: NOP

:0002_jump
wait 0
01E3: show_text_1number_styled GXT 'NUMBER' number 666 time 5000 style 1
// 666 number will appear if script works
0002: jump @0002_jump      // It jumps to specific offset.
@0002_jump_example (label) is just negative INT number, offset of label
compiled SCM file.
```

0003 - SET_CAM_SHAKE

Sanny Builder opcodes.txt: 0003: shake_camera 40

Sanny Builder SASCM.INI: 0003=1,shake_camera %1d%

Sanny Builder keyword: shake_camera (shake_camera 40)

Native name: SET_CAM_SHAKE

Description: This opcode shakes a camera

Opcode definition:

```
void set_cam_shake (int force)
```

Parameters:

1) Passed: integer, force of shaking camera. Time of shaking camera depends on this value.

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLEO }
0000: NOP

:0003_shake_camera
wait 0
if
0AB0: key_pressed 48 // 0 on left side of keyboard to press
else_jump @0003_shake_camera
0003: shake_camera force 2000
jump @0003_shake_camera
```

Watch a video presenting opcode:

0004 - set_global_int_variable

Sanny Builder opcodes.txt: 0004: \$CUSTOM_TOURNAMENT_FLAG = 0

Sanny Builder SASCM.INI: 0004=2,%1d% = %2d%

Sanny Builder short use: \$CUSTOM_TOURNAMENT_FLAG = 0

Description: Math opcode that assigns value of 4-byte size to variable. Should be used to assign integer value of global variable.

Opcode definition:

```
void set_global_int_variable (int &Global_variable, int Value);
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, value to assign

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0004_set_global_int_variable
wait 0
0004: $8201 = 92 // Global variable will get INT value
01E4: show_text_1number_lowpriority GXT 'NUMBER' number $8201 time 2000
flag 1 // Number 92 will appear
jump @0004_set_global_int_variable
```

0005 - set_global_float_variable

Sanny Builder opcodes.txt: 0005: \$166 = 292.33

Sanny Builder SASCM.INI: 0005=2,%1d% = %2d%

Sanny Builder short use: \$166 = 292.33

Description: Math opcode. It assigns value of 4-byte size to variable. Used to assign a float value a global variable.

Opcode definition:

```
void set_global_float_variable (float &Global_variable, float
Value);
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, value to assign

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04 installed.
0000: NOP

:0005_set_global_float_variable
wait 0
0005: $TEMPVAR_FLOAT_1 = 783.5 // Global variable will get FLOAT value
0AD0: show_formatted_text_lowpriority "tempvar: %f" time 2000 $TEMPVAR_FLOAT_1
jump @0005_set_global_float_variable
```

0006 - set_local_int_variable

Sanny Builder opcodes.txt: 0006: 0@ = -1

Sanny Builder SASCM.INI: 0006=2,%1d% = %2d%

Sanny Builder short use: 0@ = -1

Description: Math opcode. It assigns value of 4-byte size to variable. Should be used to assign integer value to a local variable.

Opcode definition:

```
void set_local_int_variable (int &Local_variable, int Value);
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, value to assign

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0006_set_local_int_variable
wait 0
0006: 0@ = 666 // 0@ local variable will get INT value
01E4: show_text_1number_lowpriority GXT 'NUMBER' number 0@ time 2000 flag 1
// 666 from local 0@ will appear
jump @0006_set_local_int_variable
```

0007 - set_local_float_variable

Sanny Builder opcodes.txt: 0007: 7@ = 0.0

Sanny Builder SASCM.INI: 0007=2,%1d% = %2d%

Sanny Builder short use: 0@ = 42.5

Description: Math opcode. It assigns value of 4-byte size to variable. Used to assign a float value to a local variable.

Opcode definition:

```
void set_local_float_variable (float &Local_variable, float Value);
```

Parameters:

- 1) Stored: float, local variable
- 2) Passed: float, value to assign

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04 installed
0000: NOP

:0007_set_local_float_variable
wait 0
0007: 0@ = 43.5
0AD0: show_formatted_text_lowpriority "local var float value: %f" time 2000 0@
jump @0007_set_local_float_variable
```

0008 - add_int_value_to_global_variable

Sanny Builder opcodes.txt: 0008: \$89 += 1

Sanny Builder SASCM.INI: 0008=2,%1d% += %2d%

Sanny Builder short use: \$89 += 1

Description: Math opcode. Adds the specified Integer value to the value stored in the Global Variable.

Opcode definition:

```
void add_int_value_to_global_variable (int &Global_variable, int Value);
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, value to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP
$89 = 0

:0008_add_int_value_to_global_int_variable
0008: $89 += 1 // add 1, incrementation once a second
01E4: show_text_1number_lowpriority GXT 'NUMBER' number $89 time 2000 flag 1
wait 1000
jump @0008_add_int_value_to_global_int_variable
```

0009 - add_float_value_to_global_variable

Sanny Builder opcodes.txt: 0009: \$TEMPVAR_FLOAT_1 += 1.741

Sanny Builder SASCM.INI: 0009=2,%1d% += %2d%

Sanny Builder short use: \$TEMPVAR_FLOAT_1 += 1.741

Description: Math opcode. Adds the specified float value to the value stored in the Global Variable.

Opcode definition:

```
void add_float_value_to_global_variable (float &Global_variable,
float Value);
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, value to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04 installed
0000: NOP

:0009_add_float_value_to_global_int_variable
wait 0
$TEMPVAR_FLOAT_1 = 10.0
0009: $TEMPVAR_FLOAT_1 += 2.5 // result = 12.5

0AD0: show_formatted_text_lowpriority "global float var value: %f" time 2000
$TEMPVAR_FLOAT_1
jump @0009_add_float_value_to_global_int_variable
```

000A - add_int_value_to_local_variable

Sanny Builder opcodes.txt: 000A: 3@ += 3000

Sanny Builder SASCM.INI: 000A=2,%1d% += %2d%

Sanny Builder short use: 3@ += 3000

Description: Math opcode. Adds the specified Integer value to the value stored in the Local Variable.

Opcode definition:

```
void add_int_value_to_local_variable (int &Local_variable, int
Value);
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, value to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:000A_add_int_value_to_local_variable
01E5: show_text_1number_highpriority GXT 'NUMBER' number 3@ time 5000 flag 1
000A: 3@ += 10 // Value is increased by 10 once a second
wait 1000
jump @000A_add_int_value_to_local_variable
```

000B - add_float_value_to_local_variable

Sanny Builder opcodes.txt: 000B: 6@ += 0.1

Sanny Builder SASCM.INI: 000B=2,%1d% += %2d%

Sanny Builder short use: 6@ += 0.1

Description: Math opcode. Adds the specified float value to the value stored in the Local Variable.

Opcode definition:

```
void add_float_value_to_local_variable (float &Local_variable,
float Value);
```

Parameters:

- 1) Stored: float, local variable
- 2) Passed: float, value to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04 installed
0000: NOP

:000B_add_float_value_to_local_variable
0AD1: show_formatted_text_highpriority "local float var value: %f" time 200 2@
000B: 2@ += 0.5 // Value is increased by 0.5 once a second
wait 1000
jump @000B_add_float_value_to_local_variable
```

000C - subtract_int_value_from_global_variable

Sanny Builder opcodes.txt: 000C: \$1020 -= 1

Sanny Builder SASCM.INI: 000C=2,%1d% -= %2d%

Sanny Builder short use: \$1020 -= 1

Description: Math opcode. Subtracts the specified integer value from the value stored in the Global Variable.

Opcode definition:

```
void subtract_int_value_from_global_variable (int &Global_variable,  
int Value);
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, value to subtract

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }  
0000: NOP  
  
:000C_subtract_int_value_from_global_variable  
wait 0  
$89 = 90  
000C: $89 -= 40 // Result = 50  
01E5: show_text_1number_highpriority GXT 'NUMBER' number $89 time 5000 flag 1  
jump @000C_subtract_int_value_from_global_variable
```

000D - subtract_float_value_from_global_variable

Sanny Builder opcodes.txt: 000D: \$TEMPVAR_Z_COORD -= 0.5

Sanny Builder SASCM.INI: 000D=2,%1d% -= %2d%

Sanny Builder short use: \$TEMPVAR_Z_COORD -= 0.5

Description: Math opcode. Subtracts the specified float value from the value stored in the Global Variable.

Opcode definition:

```
void subtract_float_value_from_global_variable (float  
&Global_variable, float Value);
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, value to subtract

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04 installed
0000: NOP

:000D_subtract_float_value_from_global_variable
wait 0
$tempvar_Float_1 = 2.3
000D: $tempvar_Float_1 -= 0.2 // Result = 2.1
0AD1: show_formatted_text_highpriority "global float var value: %f" time 200
$tempvar_Float_1
jump @000D subtract float value from global variable
```

000E - subtract_int_value_from_local_variable

Sanny Builder opcodes.txt: 000E: 0@ -= 1

Sanny Builder SASCM.INI: 000E=2,%1d% -= %2d%

Sanny Builder short use: 0@ -= 1

Description: Math opcode. Subtracts the specified integer value from the value stored in the Local Variable.

Opcode definition:

```
void subtract_int_value_from_local_variable (int &Local_variable,
int Value);
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, value to subtract

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:000E_subtract_int_value_from_local_variable
wait 0
0@ = 47
000E: 0@ -= 5 // Result = 42
01E5: show_text_1number_highpriority GXT 'NUMBER' number 0@ time 5000 flag 1
jump @000E_subtract_int_value_from_local_variable
```

000F - subtract_float_value_from_local_variable

Sanny Builder opcodes.txt: 000F: 692@ -= 8.0

Sanny Builder SASCM.INI: 000F=2,%1d% -= %2d%

Sanny Builder short use: 692@ -= 8.0

Description: Math opcode. Subtracts the specified float value from the value stored in the Local Variable.

Opcode definition:

```
void subtract_float_value_from_local_variable (float  
&Local_variable, float Value);
```

Parameters:

- 1) Stored: float, local variable
- 2) Passed: float, value to subtract

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }  
0000: NOP  
  
:000F_subtract_float_value_from_local_variable  
wait 0  
0@ = 47.5  
000F: 0@ -= 3.0      // Result = 44.5  
0AD1: show_formatted_text_highpriority "local float var value: %f" time 200 0@  
jump @000F_subtract_float_value_from_local_variable
```

0010 - multiply_by_int_value_in_global_variable

Sanny Builder opcodes.txt: 0010: \$GS_GANG_CASH *= 100

Sanny Builder SASCM.INI: 0010: \$GS_GANG_CASH *= 100

Sanny Builder short use: \$GS_GANG_CASH *= 100

Description: Math opcode. Multiplies the value stored in the Global Variable by the specified Integer value.

Opcode definition:

```
void multiply_by_int_value_in_global_variable (int  
&Global_variable, int Value);
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0010_multiply_by_int_value_in_global_variable
wait 0
$89 = 5
0010: $89 *= 10 // Result = 50
01E5: show_text_1number_highpriority GXT 'NUMBER' number $89 time 5000 flag 1
jump @0010_multiply_by_int_value_in_global_variable
```

0011 - multiply_by_float_value_in_global_variable

Sanny Builder opcodes.txt: 0011: \$HJ_TEMP_FLOAT *= 100.0

Sanny Builder SASCM.INI: 0011=2,%1d% *= %2d%

Sanny Builder short use: \$HJ_TEMP_FLOAT *= 100.0

Description: Math opcode. Multiplies the value stored in the Global Variable by the specified Float value.

Opcode definition:

```
void multiply_by_float_value_in_global_variable (float
&Global_variable, float Value);
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0011_multiply_by_float_value_in_global_variable
wait 0
$tempvar_Float_1 = 5.0
0011: $tempvar_Float_1 *= 100.0 // Result = 500.0
0AD1: show_formatted_text_highpriority "global float var value: %f" time 200
$tempvar_Float_1
iump @0011 multiply by float value in global variable
```

0012 - multiply_by_int_value_in_local_variable

Sanny Builder opcodes.txt: 0012: 22@ *= -1

Sanny Builder SASCM.INI: 0012=2,%1d% *= %2d%

Sanny Builder short use: 22@ *= -1

Description: Math opcode. Multiplies the value stored in the Local Variable by the specified Integer value.

Opcode definition:

```
void multiply_by_int_value_in_local_variable (int &Local_variable,
int Value);
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0012_multiply_by_int_value_in_local_variable
wait 0
0@ = 5
0012: 0@ *= 2           // Result = 10
01E5: show_text_1number_highpriority GXT 'NUMBER' number 0@ time 5000 flag 1
jump @0012_multiply_by_int_value_in_local_variable
```

0013 - multiply_by_float_value_in_local_variable

Sanny Builder opcodes.txt: 0013: 17@ *= 9.8

Sanny Builder SASCM.INI: 0013=2,%1d% *= %2d%

Sanny Builder short use: 17@ *= 9.8

Description: Math opcode. Multiplies the value stored in the Local Variable by the specified Float value.

Opcode definition:

```
void multiply_by_float_value_in_local_variable (float
&Local_variable, float Value);
```

Parameters:

- 1) Stored: float, local variable
- 2) Passed: float, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0013_multiply_by_float_value_in_local_variable
wait 0
0@ = 2.5
0013: 0@ *= 2.0 // Result = 5.0
0AD1: show_formatted_text_highpriority "local var float value: %f" time 200 0@
jump @0013 multiplv bv float value in local variable
```

0014 - divide_by_int_value_in_global_variable

Sanny Builder opcodes.txt: 0014: \$HJ_TWOWHEELS_TIME /= 1000

Sanny Builder SASCM.INI: 0014=2,%1d% /= %2d%

Sanny Builder short use: \$HJ_TWOWHEELS_TIME /= 1000

Description: Math opcode. Divides the value stored in the Global Variable by the specified Integer value.

Opcode definition:

```
void divide_by_int_value_in_global_variable (int &Global_variable,
int Value);
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0014_divide_by_int_value_in_global_variable
wait 0
$89 = 10
0014: $89 /= 5 // Result = 2

01E5: show_text_1number_highpriority GXT 'NUMBER' number $89 time 5000 flag 1
jump @0014 divide by int value in global variable
```

0015 - divide_by_float_value_in_global_variable

Sanny Builder opcodes.txt: 0015: \$EXPORT_PRICE_HEALTH_MULTIPLIER /= 1000

Sanny Builder SASCM.INI: 0015=2,%1d% /= %2d%

Sanny Builder short use: \$EXPORT_PRICE_HEALTH_MULTIPLIER /= 1000

Description: Math opcode. Divides the value stored in the Global Variable by the specified Float value.

Opcode definition:

```
void divide_by_float_value_in_global_variable (float  
&Global_variable, float Value);
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }  
// Example requires CLE04  
0000: NOP  
  
:0015_divide_by_float_value_in_global_variable  
wait 0  
$tempvar_Float_1 = 5.0  
0015: $tempvar_Float_1 /= 2.5 // Result = 2.0  
  
0AD1: show_formatted_text_highpriority "global var float value: %f" time 200  
$tempvar_Float_1  
jump @0015 divide hv float value in global variable
```

0016 - divide_by_int_value_in_local_variable

Sanny Builder opcodes.txt: 0016: 4@ /= 2

Sanny Builder SASCM.INI: 0016=2,%1d% /= %2d%

Sanny Builder short use: 4@ /= 2

Description: Math opcode. Divides the value stored in the Local Variable by the specified Integer value.

Opcode definition:

```
void divide_by_int_value_in_local_variable (int &Local_variable, int  
Value);
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

:0016_divide_by_int_value_in_local_variable
wait 0
4@ = 8
0016: 4@ /= 2 // Result = 4

01E5: show_text_1number_highpriority GXT 'NUMBER' number 4@ time 5000 flag 1
jump @0016_divide_by_int_value_in_local_variable
```

0017 - divide_by_float_value_in_local_variable

Sanny Builder opcodes.txt: 0017: 14@ /= 1000.0

Sanny Builder SASCM.INI: 0017=2,%1d% /= %2d%

Sanny Builder short use: 14@ /= 1000.0

Description: Math opcode. Divides the value stored in the Local Variable by the specified Float value.

Opcode definition:

```
void divide_by_float_value_in_local_variable (float &Local_variable,
float Value);
```

Parameters:

- 1) Stored: float, local variable
- 2) Passed: float, value to multiply by

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0017_divide_by_float_value_in_local_variable
wait 0
14@ = 6000.0
0017: 14@ /= 1000.0 // Result = 6.0

0AD1: show_formatted_text_highpriority "local var float value: %f" time 200
14@
jump @0017_divide_by_float_value_in_local_variable
```

0018 - is_int_global_variable_greater_than_value

Sanny Builder opcodes.txt: 0018: \$CATALINA_TOTAL_PASSED_MISSIONS > 2

Sanny Builder SASCM.INI: 0018=2, %1d% > %2d%

Sanny Builder short use: \$CATALINA_TOTAL_PASSED_MISSEIONS > 2

Description: Math opcode. Checks if Integer value stored in Global variable is greater than Integer value passed in second parameter.

Opcode definition:

```
bool is_int_global_variable_greater_than_value (int Global_variable,
int Value)
```

Parameters:

- 1) Passed: Integer, Global variable, supposed as greater
- 2) Passed: Integer, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0018_is_int_global_variable_greater_than_value
wait 0
$98 = 530

if
0018: $98 > 150
then
0AD1: show_formatted_text_highpriority "Global variable is greater than
value 150" time 2000 // It will appear
else
0AD1: show_formatted_text_highpriority "Global variable is lower than
value 150" time 2000
end
jump @0018_is_int_global_variable_greater_than_value
```

0019 - is_int_local_variable_greater_than_value

Sanny Builder opcodes.txt: 0019: 0@ > 0

Sanny Builder SASCM.INI: 0019=2, %1d% > %2d%

Sanny Builder short use: 0@ > 0

Description: Math opcode. Checks if Integer value stored in Local variable is greater than Integer value passed in second parameter.

Opcode definition:

```
bool is_int_local_variable_greater_than_value (int Local_variable,
int Value);
```

Parameters:

- 1) Passed: Integer, Local variable, supposed as greater
- 2) Passed: Integer, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0019_is_int_local_variable_greater_than_value
wait 0
0@ = 101

if
0019: 0@ > 100
then
0AD1: show_formatted_text_highpriority "Local variable is greater than
value 100" time 2000 // It will appear
else
0AD1: show_formatted_text_highpriority "Local variable is lower than
value 100" time 2000
end
iump @0019 is int local variable greater than value
```

001A - is_int_value_greater_than_global_variable

Sanny Builder opcodes.txt: 001A: 10 > \$SYNDICATE_TOTAL_PASSED_MISSIONS

Sanny Builder SASCM.INI: 001A=2, %1d% > %2d%

Sanny Builder short use: 10 > \$SYNDICATE_TOTAL_PASSED_MISSIONS

Description: Math opcode. Checks if first parameter, given Integer value is greater than the value stored in Global variable - returns true if value stored in Global variable is lower than given value.

Opcode definition:

```
bool is_int_value_greater_than_global_variable (int Value , int
```

Parameters:

- 1) Passed: Integer, value supposed as greater
- 2) Passed: Integer, Global variable, variable to compare value

Returns true or false? Yes.**Example in Sanny Builder:**

```

{$CLE0}
// Example requires CLE04
0000: NOP

:001A_is_int_value_greater_than_global_variable
wait 0
$98 = 30

    if
    001A:    35 > $98
    then
    0AD1: show_formatted_text_highpriority "Global variable is lower than
value 35" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "Global variable is greater than
value 35" time 2000
    end
jump @001A_is_int_value_greater_than_global_variable

```

001B - is_int_value_greater_than_local_variable**Sanny Builder opcodes.txt:** 001B: 3 > 20@**Sanny Builder SASCM.INI:** 001b=2, %1d% > %2d%**Sanny Builder short use:** 3 > 20@

Description: Math opcode. Checks if first parameter, given Integer value is greater than the value stored in Local variable - returns true if value stored in Local variable is lower than given value.

Opcode definition:

```
bool is_int_value_greater_than_local_variable (int Value , int
```

Parameters:

- 1) Passed: Integer, value supposed as greater
- 2) Passed: Integer, Local variable, variable to compare value with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:001B_is_int_value_greater_than_local_variable
wait 0
20@ = 1

    if
    001B: 3 > 20@
    then
    0AD1: show_formatted_text_highpriority "INT Value from above is greater
than value from local variable" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT Value from above is lower
than value from local variable" time 2000
    end

jump @001B_is_int_value_greater_than_local_variable
```

001C - is_int_global_variable_greater_than_global_variable

Sanny Builder opcodes.txt: 001C: \$CURRENT_MONTH_DAY >
\$GYM_MONTH_DAY_WHEN_LIMIT_REACHED // (int)

Sanny Builder SASCM.INI: 001c=2, %1d% > %2d% ; (int)

Sanny Builder short use: \$CURRENT_MONTH_DAY >
\$GYM_MONTH_DAY_WHEN_LIMIT_REACHED

Variable types must be declared before.

Description: Math opcode. Checks if first global integer variable is greater than the second global variable and returns true or false.

Opcode definition:

```
bool is_int_global_variable_greater_than_global_variable (int
Global_variable , int Global_variable)
```

Parameters:

- 1) Passed: Integer, Global variable, supposed as greater
- 2) Passed: Integer, Global variable, variable to compare first

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:001C_is_int_global_variable_greater_than_global_variable
wait 0
$89 = 7
$90 = 5

    if
    001C:  $89 > $90 // (int)
    then
    0AD1: show_formatted_text_highpriority "INT first variable is greater
than the second" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT first variable is lower than
the second" time 2000
    end

// 001C is int global variable greater than global variable
```

001D - is_int_local_variable_greater_than_local_variable

Sanny Builder opcodes.txt: 001D: 27@ > 33@ // (int)

Sanny Builder SASCM.INI: 001d=2, %1d% > %2d% ; (int)

Sanny Builder short use: 27@ > 33@

Variable types must be declared before.

Description: Math opcode. Checks if first local integer variable is greater than the second local variable and returns true or false.

Opcode definition:

```
bool is_int_local_variable_greater_than_local_variable (int Local_variable
, int Local_variable)
```

Parameters:

- 1) Passed: Integer, Local variable, supposed as greater
- 2) Passed: Integer, Local variable, variable to compare first

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:001D_is_int_local_variable_greater_than_local_variable
wait 0
0@ = 50 // bigger
1@ = 20

if
001D: 0@ > 1@ // (int)
then
0AD1: show_formatted_text_highpriority "INT first variable is greater
than the second" time 2000 // It will appear
else
0AD1: show_fomatted_text_highpriority "INT first variable is lower than
the second" time 2000
end

//--- 001D_is_int_local_variable_greater_than_local_variable
```

001E - is_int_global_variable_greater_than_local_variable

Sanny Builder opcodes.txt: 001E: \$CURRENT_TIME_IN_MS2 > 3@ // (int)

Sanny Builder SASCM.INI: 001E=2, %1d% > %2d% ; (int)

Sanny Builder short use: \$CURRENT_TIME_IN_MS2 > 3@

Variable types must be declared before.

Description: Math opcode. Checks if the integer value of first global variable is greater than the value of second local variable and returns true or false.

Opcode definition:

```
bool is_int_global_variable_greater_than_local_variable (int
Global_variable , int Local_variable)
```

Parameters:

- 1) Passed: Integer, Global variable, supposed as greater
- 2) Passed: Integer, Local variable, variable to compare first

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:001E_is_int_global_variable_greater_than_local_variable
wait 0
$89 = 60      // bigger
0@ = 55       // smaller

    if
    001E:  $89 > 0@ // (int)
    then
    0AD1: show_formatted_text_highpriority "INT first variable is greater
than the second" time 2000 // It will appear
    else
    0AD1: show_fomatted_text_highpriority "INT first variable is lower than
the second" time 2000
    end

// 001F is int global variable greater than local variable
```

001F - is_int_local_variable_greater_than_global_variable

Sanny Builder opcodes.txt: 001F: 9@ > \$GIRL_PROGRESS[0] // (int)

Sanny Builder SASCM.INI: 001f=2, %1d% > %2d% ; (int)

Sanny Builder short use: 9@ > \$GIRL_PROGRESS[0]

Variable types must be declared before.

Description: Math opcode. Checks if the integer value of first local variable is greater than the value of second local variable and returns true or false.

Opcode definition:

```
bool is_int_local_variable_greater_than_global_variable (int
Local_variable , int Global_variable)
```

Parameters:

- 1) Passed: Integer, Local variable, supposed as greater
- 2) Passed: Integer, Global variable, variable to compare first

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:001F_is_int_local_variable_greater_than_global_variable
wait 0
0@ = 60 // bigger
$89 = 55 // smaller

if
001F: 0@ > $89 // (int)
then
0AD1: show_formatted_text_highpriority "INT first variable is greater
than the second" time 2000 // It will appear
else
0AD1: show_fomatted_text_highpriority "INT first variable is lower than
the second" time 2000
end

// --- :001F_is_int_local_variable_greater_than_global_variable
```

0020 - is_float_global_variable_greater_than_value

Sanny Builder opcodes.txt: 0020: \$HJ_TWOWHEELS_DISTANCE_FLOAT > 0.0

Sanny Builder SASCM.INI: 0020=2, %1d% > %2d%

Sanny Builder short use: \$HJ_TWOWHEELS_DISTANCE_FLOAT > 0.0

Description: Math opcode. Checks if Float value stored in Global variable is greater than Float value passed in second parameter.

Opcode definition:

```
bool is_float_global_variable_greater_than_value (float Global_variable,
float Value)
```

Parameters:

- 1) Passed: Float, Global variable, supposed as greater
- 2) Passed: Float, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0020_is_float_global_variable_greater_than_value
wait 0
$tempvar_Float_1 = 2.5

    if
    0020: $tempvar_Float_1 > 2.2
    then
    0AD1: show_formatted_text_highpriority "FLOAT first variable is greater
than the value" time 2000 // It will appear
    else
    0AD1: show_fomatted_text_highpriority "FLOAT first variable is lower
than the value" time 2000
    end

jump 0020_is_float_global_variable_greater_than_value
```

0021 - is_float_local_variable_greater_than_value

Sanny Builder opcodes.txt: 0021: 26@ > 64.0

Sanny Builder SASCM.INI: 0021=2, %1d% > %2d%

Sanny Builder short use: 26@ > 64.0

Description: Math opcode. Checks if Float value stored in Local variable is greater than Float value passed in second parameter.

Opcode definition:

```
bool is_float_local_variable_greater_than_value (float Local_variable,
float Value)
```

Parameters:

- 1) Passed: Float, Local variable, supposed as greater
- 2) Passed: Float, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0021_is_float_local_variable_greater_than_value
wait 0
26@ = 70.0

    if
    0021: 26@ > 64.0
    then
    0AD1: show_formatted_text_highpriority "FLOAT first variable is greater
than the value" time 2000 // It will appear
    else
    0AD1: show_fomatted_text_highpriority "FLOAT first variable is lower
than the value" time 2000
    end

jump 0021_is_float_local_variable_greater_than_value
```

0022 - is_float_value_greater_than_global_variable

Sanny Builder opcodes.txt: 0022: -180.0 > \$1316

Sanny Builder SASCM.INI: 0022=2, %1d% > %2d%

Sanny Builder short use: -180.0 > \$1316

Description: Math opcode. Checks if Float value is greater than the value stored in Global variable.

Opcode definition:

```
bool is_float_value_greater_than_global_variable (float Value, float
Global_variable)
```

Parameters:

- 1) Passed: Float, value supposed as greater
- 2) Passed: Float, Global variable to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0022_is_float_value_greater_than_global_variable
wait 0
$tempvar_Float_1 = 70.0

    if
    0022: 80.0 > $tempvar_Float_1
    then
    0AD1: show_formatted_text_highpriority "FLOAT checked value is greater
than the value in global variable" time 2000 // It will appear
    else
    0AD1: show_fomatted_text_highpriority "FLOAT checked value is lower than
the value in global variable" 2000
    end

jump 0023_is_float_value_greater_than_global_variable
```

0023 - is_float_value_greater_than_local_variable

Sanny Builder opcodes.txt: 0023: 0.0 > 7@

Sanny Builder SASCM.INI: 0023=2, %1d% > %2d%

Sanny Builder short use: 0.0 > 7@

Description: Math opcode. Checks if Float value is greater than the value stored in Local variable.

Opcode definition:

```
bool is_float_value_greater_than_local_variable (float Value, float
Local_variable)
```

Parameters:

- 1) Passed: Float, value supposed as greater
- 2) Passed: Float, Local variable to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0023_is_float_value_greater_than_local_variable
wait 0
0@ = 20.0

    if
    0023: 30.5 > 0@
    then
    0AD1: show_formatted_text_highpriority "FLOAT checked value is greater
than the value in local variable" time 2000 // It will appear
    else
    0AD1: show_fomatted_text_highpriority "FLOAT checked value is lower than
the value in local variable" 2000
    end

jump 0023_is_float_value_greater_than_local_variable
```

0024 - is_float_global_variable_greater_than_global_variable

Sanny Builder opcodes.txt: 0024: \$HJ_CAR_Z > \$HJ_CAR_Z_MAX // (float)

Sanny Builder SASCM.INI: 0024=2, %1d% > %2d% ; (float)

Sanny Builder short use: \$HJ_CAR_Z > \$HJ_CAR_Z_MAX

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of first global variable is greater than the value of second global variable and returns true or false.

Opcode definition:

```
bool is_float_global_variable_greater_than_global_variable (float
Global_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Global variable, supposed as greater
- 2) Passed: Float, Global variable, variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0024_is_float_value_greater_than_local_variable
wait 0
$tempvar_Float_1 = 30.0
$tempvar_Float_2 = 25.0

    if
    0024: $tempvar_Float_1 > $tempvar_Float_2 // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
global var is greater than the value in second global variable" time 2000
// It will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
global var is lower than the value in second global variable" 2000
    end

jump 0024_is_float_value_greater_than_local_variable
```

0025 - is_float_local_variable_greater_than_local_variable

Sanny Builder opcodes.txt: 0025: 3@ > 6@ // (float)

Sanny Builder SASCM.INI: 0025=2, %1d% > %2d% ; (float)

Sanny Builder short use: 3@ > 6@

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of first local variable is greater than the value of second local variable and returns true or false.

Opcode definition:

```
bool is_float_local_variable_greater_than_local_variable (float
Local_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Local variable, supposed as greater
- 2) Passed: Float, Local variable, variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0025_is_float_local_variable_greater_than_local_variable
wait 0
0@ = 4.4
1@ = 2.2

    if
    0025: 0@ > 1@ // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
local var is greater than the value in second local variable" time 2000 //
It will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
local var is lower than the value in second local variable" 2000
    end

// 0025_is_float_local_variable_greater_than_local_variable
```

0026 - is_float_global_variable_greater_than_local_variable

Sanny Builder opcodes.txt: 0026: \$TEMPVAR_FLOAT_1 > 513@(227@,10f) //
(float)

Sanny Builder SASCM.INI: 0026=2, %1d% > %2d% ; (float)

Sanny Builder short use: \$TEMPVAR_FLOAT_1 > 513@(227@,10f)

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of first global variable is greater than the value of second local variable and returns true or false.

Opcode definition:

```
bool is_float_global_variable_greater_than_local_variable (float
Global_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Global variable, supposed as greater
- 2) Passed: Float, Local variable, variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0026_is_float_global_variable_greater_than_local_variable
wait 0
$TEMPVAR_FLOAT_1 = 5.5 // none of variables is greater...
0@ = 5.5

if
0026: $TEMPVAR_FLOAT_1 > 0@ // (float)
then
0AD1: show_formatted_text_highpriority "FLOAT checked value of first
global var is greater than the value in second local variable" time 2000
else
0AD1: show_formatted_text_highpriority "FLOAT checked value of first
global var is not greater than the value in second local variable" 2000 //
It will appear
end

// :0026_is_float_global_variable_greater_than_local_variable
```

0027 - is_float_local_variable_greater_than_global_variable

Sanny Builder opcodes.txt: 0027: 513@(227@,10f) > \$TEMPVAR_FLOAT_2 // (float)

Sanny Builder SASCM.INI: 0027=2, %1d% > %2d% ; (float)

Sanny Builder short use: 513@(227@,10f) > \$TEMPVAR_FLOAT_2

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of first local variable is greater than the value of second global variable and returns true or false.

Opcode definition:

```
bool is_float_local_variable_greater_than_global_variable (float
Local_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Local variable, supposed as greater
- 2) Passed: Float, Global variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0027_is_float_local_variable_greater_than_global_variable
wait 0
0@ = 5.5 // none of variables is greater...
$TEMPVAR_FLOAT_1 = 5.5

    if
    0027: 0@ > $TEMPVAR_FLOAT_1 // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
local var is greater than the value in second global variable" time 2000
    else
    0AD1: show_formatted_text_highpriority "FLOAT checked value of first
local var is not greater than the value in second global variable" 2000 //
It will appear
    end

// 0027_is_float_local_variable_greater_than_global_variable
```

0028 - is_int_global_variable_greater_or_equal_to_value

Sanny Builder opcodes.txt: 0028: \$5283 >= 180000

Sanny Builder SASCM.INI: 0028=2, %1d% >= %2d%

Sanny Builder short use: \$5283 >= 180000

Description: Math opcode. Checks if Integer value stored in Global variable is greater or equal to Integer value passed in second parameter.

Opcode definition:

```
bool is_int_global_variable_greater_or_equal_to_value (int
Global_variable, int Value)
```

Parameters:

- 1) Passed: Integer, Global variable, supposed as greater or equal
- 2) Passed: Integer, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0028_is_int_global_variable_greater_or_equal_to_value
wait 0
$89 = 67

    if
    0028:  $89 >= 67
    then
    0AD1: show_formatted_text_highpriority "INT value of global variable is
greater or equal to value above" time 2000
    else
    0AD1: show_formatted_text_highpriority "INT value of global variable is
lower than value above" 2000 // It will appear
    end

    jump 0029_is_int_global_variable_greater_or_equal_to_value
```

0029 - is_int_local_variable_greater_or_equal_to_value

Sanny Builder opcodes.txt: 0029: 17@ >= 4

Sanny Builder SASCM.INI: 0029=2, %1d% >= %2d%

Sanny Builder short use: 17@ >= 4

Description: Math opcode. Checks if Integer value stored in Local variable is greater or equal to Integer value passed in second parameter.

Opcode definition:

```
bool is_int_local_variable_greater_or_equal_to_value (int Local_variable,
int Value)
```

Parameters:

- 1) Passed: Integer, Local variable, supposed as greater or equal
- 2) Passed: Integer, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0029_is_int_local_variable_greater_or_equal_to_value
wait 0
0@ = 6

    if
    0029: 0@ >= 4
    then
    0AD1: show_formatted_text_highpriority "INT value of local variable is
greater or equal to value above" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT value of local variable is
lower than value above" 2000
    end

jump 0029_is_int_local_variable_greater_or_equal_to_value
```

002A - is_int_value_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 002A: 0 >= \$GIRL_PROGRESS[4]

Sanny Builder SASCM.INI: 002a=2, %1d% >= %2d%

Sanny Builder short use: 0 >= \$GIRL_PROGRESS[4]

Description: Math opcode. Checks if passed Integer value is greater or equal to Integer value stored in Global variable.

Opcode definition:

```
bool is_int_value_greater_or_equal_to_global_variable (int Value, int
Global_variable)
```

Parameters:

- 1) Passed: Integer, value supposed as greater or equal
- 2) Passed: Integer, Global variable to compare first parameter

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002A_is_int_value_greater_or_equal_to_global_variable
wait 0
$89 = 64

    if
    002A: 70 >= $89
    then
    0AD1: show_formatted_text_highpriority "INT value from above of is
greater or equal to value of global variable" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT value from above is lower
than of global variable" 2000
    end

jump @002A_is_int_value_greater_or_equal_to_global_variable
```

002B - is_int_value_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 002B: 3000 >= 8@

Sanny Builder SASCM.INI: 002B: 3000 >= 8@

Sanny Builder short use: 3000 >= 8@

Description: Math opcode. Checks if passed Integer value is greater or equal to Integer value stored in Local variable.

Opcode definition:

```
bool is_int_value_greater_or_equal_to_local_variable (int Value, int
Local_variable)
```

Parameters:

- 1) Passed: Integer, value supposed as greater or equal
- 2) Passed: Integer, Global variable to compare first parameter

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002B_is_int_value_greater_or_equal_to_local_variable
wait 0
0@ = 2000

    if
    002B: 3000 >= 0@
    then
    0AD1: show_formatted_text_highpriority "INT value from above of is
greater or equal to value of local variable" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT value from above is lower
than value of local variable" 2000
    end

jump 002B_is_int_value_greater_or_equal_to_local_variable
```

002C - is_int_global_variable_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 002C: \$SAVE_PICKUPS_INDEX >=
\$TOTAL_AVAILABLE_SAVE_PICKUPS // (int)

Sanny Builder SASCM.INI: 002c=2, %1d% >= %2d% ; (int)

Sanny Builder short use: \$SAVE_PICKUPS_INDEX >=
\$TOTAL_AVAILABLE_SAVE_PICKUPS

Variable types must be declared before.

Description: Math opcode. Checks if Integer value of the first Global variable is greater or equal to the Integer value stored in second Global variable.

Opcode definition:

```
bool is_int_global_variable_greater_or_equal_to_global_variable (int
Global_variable, int Global_variable)
```

Parameters:

- 1) Passed: Integer, Global variable supposed as greater or equal
- 2) Passed: Integer, Global variable to compare first parameter

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002C_is_int_global_variable_greater_or_equal_to_global_variable
wait 0
$89 = 30
$90 = 25

if
002C: $89 >= $90 // (int)
then
0AD1: show_formatted_text_highpriority "INT value of first global
variable of is greater or equal to the value of second global variable" time
2000 // It will appear
else
0AD1: show_formatted_text_highpriority "INT value of first global
variable is lower than the value of second global variable" 2000
end

jump @002C_is_int_global_variable_greater_or_equal_to_global_variable
```

002D - is_int_local_variable_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 002D: 43@ >= 271@ // (int)

Sanny Builder SASCM.INI: 002d=2, %1d% >= %2d% ; (int)

Sanny Builder short use: 43@ >= 271@

Variable types must be declared before.

Description: Math opcode. Checks if Integer value of the first Local variable is greater or equal to the Integer value stored in second Local variable.

Opcode definition:

```
bool is_int_local_variable_greater_or_equal_to_local_variable (int
Local_variable, int Local_variable)
```

Parameters:

- 1) Passed: Integer, Local variable supposed as greater or equal
- 2) Passed: Integer, local variable to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002D_is_int_local_variable_greater_or_equal_to_local_variable
wait 0
0@ = 50      // bigger or equal
1@ = 20      // smaller

    if
    002D: 0@ >= 1@ // (int)
    then
    0AD1: show_formatted_text_highpriority "INT value of first local
variable of is greater or equal to the value of second local variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT value of first local
variable is lower than the value of second local variable" 2000
    end

jump @002D_is_int_local_variable_greater_or_equal_to_local_variable
```

002E - is_int_global_variable_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 002E: \$DIALOG_ARRAY_SIZE >= 131@ // (int)

Sanny Builder SASCM.INI: 002e=2, %1d% >= %2d% ; (int)

Sanny Builder short use: \$DIALOG_ARRAY_SIZE >= 131@

Variable types must be declared before.

Description: Math opcode. Checks if Integer value of the first Global variable is greater or equal to the Integer value stored in second Local variable.

Opcode definition:

```
bool is_int_global_variable_greater_or_equal_to_local_variable (int
Global_variable, int Local_variable)
```

Parameters:

- 1) Passed: Integer, Global variable supposed as greater or equal
- 2) Passed: Integer, Local variable to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002E_is_int_global_variable_greater_or_equal_to_local_variable
wait 0
$89 = 50      // bigger or equal
1@ = 20       // smaller

    if
    002E:  $89 >= 1@ // (int)
    then
    0AD1:  show_formatted_text_highpriority "INT value of first global
variable of is greater or equal to the value of second local variable" time
2000 // It will appear
    else
    0AD1:  show_formatted_text_highpriority "INT value of first global
variable is lower than the value of second local variable" 2000
    end

jump @002F_is_int_local_variable_greater_or_equal_to_global_variable
```

002F - is_int_local_variable_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 002F: 1@ >= \$1264(\$1288,2i) // (int)

Sanny Builder SASCM.INI: 002F=2, %1d% >= %2d% ; (int)

Sanny Builder short use: 1@ >= \$1264(\$1288,2i)

Variable types must be declared before.

Description: Math opcode. Checks if Integer value of the first Local variable is greater or equal to the Integer value stored in second Global variable.

Opcode definition:

```
bool is_int_local_variable_greater_or_equal_to_global_variable (int
Local_variable, int Global_variable)
```

Parameters:

- 1) Passed: Integer, Local variable supposed as greater or equal
- 2) Passed: Integer, Global variable to compare first parameter

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:002F_is_int_local_variable_greater_or_equal_to_global_variable
wait 0
0@ = 50      // bigger
$89 = 20      // smaller

    if
    002F:  0@ >= $89 // (int)
    then
    0AD1: show_formatted_text_highpriority "INT value of first local
variable of is greater or equal to the value of second global variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "INT value of first local
variable is lower than the value of second global variable" 2000
    end

jump @002F_is_int_local_variable_greater_or_equal_to_global_variable
```

0030 - is_float_global_variable_greater_or_equal_to_value

Sanny Builder opcodes.txt: 0030: \$STAT_PERCENTAGE_COMPLETED >= 100.0

Sanny Builder SASCM.INI: 0030=2, %1d% >= %2d%

Sanny Builder short use: \$STAT_PERCENTAGE_COMPLETED >= 100.0

Description: Math opcode. Checks if Float value of the first Global variable is greater or equal to the Float value passed in second parameter.

Opcode definition:

```
bool is_float_global_variable_greater_or_equal_to_value (float
Global_variable, float Value)
```

Parameters:

- 1) Passed: Float, Global variable supposed as greater or equal
- 2) Passed: Float, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0030_is_float_global_variable_greater_or_equal_to_value
wait 0
$89 = 120.0

    if
    0030: $89 >= 100.0
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of first global
variable of is greater or equal to the value" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of first global
variable is lower than the value" 2000
    end

jump @0030_is_float_global_variable_greater_or_equal_to_value
```

0031 - is_float_local_variable_greater_or_equal_to_value

Sanny Builder opcodes.txt: 0031: 42@ >= 0.05

Sanny Builder SASCM.INI: 0031=2, %1d% >= %2d%

Sanny Builder short use: 42@ >= 0.05

Description: Math opcode. Checks if Float value of the first Local variable is greater or equal to the Float value passed in second parameter.

Opcode definition:

```
bool is_float_local_variable_greater_or_equal_to_value (float
Local_variable, float Value)
```

Parameters:

- 1) Passed: Float, Local variable supposed as greater or equal
- 2) Passed: Float, value to compare first parameter with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0031_is_float_local_variable_greater_or_equal_to_value
wait 0
0@ = 120.0

    if
    0031: 0@ >= 0.05
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of first local
variable of is greater or equal to the value" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of first local
variable is lower than the value" 2000
    end

jump @0031_is_float_local_variable_greater_or_equal_to_value
```

0032 - is_float_value_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 0032: 8.0 >= \$5925[0]

Sanny Builder SASCM.INI: 0032=2, %1d% >= %2d%

Sanny Builder short use: 8.0 >= \$5925[0]

Description: Math opcode. Checks if specified Float value is greater or equal to the Float value stored in Global variable.

Opcode definition:

```
bool is_float_value_greater_or_equal_to_global_variable (float Value,
float Global_variable)
```

Parameters:

- 1) Passed: Float, value supposed as greater or equal
- 2) Passed: Float, Global variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }  
// Example requires CLE04  
0000: NOP  
  
:0032_is_float_value_greater_or_equal_to_global_variable  
wait 0  
$tempvar_Float_1 = 2.1  
  
    if  
    0032: 8.0 >= $tempvar_Float_1  
    then  
    0AD1: show_formatted_text_highpriority "specified FLOAT value is greater  
or equal to the value of global variable" time 2000 // It will appear  
    else  
    0AD1: show_formatted_text_highpriority "specified FLOAT value is lower  
than value of global variable" 2000  
    end  
  
jump @0032_is_float_value_greater_or_equal_to_global_variable
```

0033 - is_float_value_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 0033: -0.05 >= 42@

Sanny Builder SASCM.INI: 0033=2, %1d% >= %2d%

Sanny Builder short use: -0.05 >= 42@

Description: Math opcode. Checks if specified Float value is greater or equal to the Float value stored in Local variable.

Opcode definition:

```
bool is_float_value_greater_or_equal_to_local_variable (float Value, float  
Local_variable)
```

Parameters:

- 1) Passed: Float, value supposed as greater or equal
- 2) Passed: Float, Local variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0033_is_float_value_greater_or_equal_to_local_variable
wait 0
0@ = 70.3

    if
    0033: 70.6 >= 0@
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value is greater
or equal to the value of local variable" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value is lower
than value of local variable" 2000
    end

jump @0033_is_float_value_greater_or_equal_to_local_variable
```

0034 - is_float_global_variable_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 0034: \$8276 >= \$8278 // (float)

Sanny Builder SASCM.INI: 0034=2, %1d% >= %2d% ; (float)

Sanny Builder short use: \$8276 >= \$8278

Variable types must be declared before.

Description: Math opcode. Checks if specified Float value of the first Global variable is greater or equal to the Float value stored in second Global variable.

Opcode definition:

```
bool is_float_value_greater_or_equal_to_local_variable (float
Global_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Global variable supposed as greater or equal
- 2) Passed: Float, Global variable variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0034_is_float_global_variable_greater_or_equal_to_global_variable
wait 0
$tempvar_Float_1 = 65.0
$tempvar_Float_2 = 50.0

    if
    0034: $tempvar_Float_1 >= $tempvar_Float_2 // (float)
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is greater or equal to the value of second global variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is lower than the value of second global variable" 2000
    end

jump @0034_is_float_global_variable_greater_or_equal_to_global_variable
```

0034 - is_float_global_variable_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 0034: \$8276 >= \$8278 // (float)

Sanny Builder SASCM.INI: 0034=2, %1d% >= %2d% ; (float)

Sanny Builder short use: \$8276 >= \$8278

Variable types must be declared before.

Description: Math opcode. Checks if specified Float value of the first Global variable is greater or equal to the Float value stored in second Global variable.

Opcode definition:

```
bool is_float_global_variable_greater_or_equal_to_global_variable
(float Global_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Global variable supposed as greater or equal
- 2) Passed: Float, Global variable variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0034_is_float_global_variable_greater_or_equal_to_global_variable
wait 0
$tempvar_Float_1 = 65.0
$tempvar_Float_2 = 50.0

    if
    0034: $tempvar_Float_1 >= $tempvar_Float_2 // (float)
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is greater or equal to the value of second global variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is lower than the value of second global variable" 2000
    end

jump @0034_is_float_global_variable_greater_or_equal_to_global_variable
```

0035 - is_float_local_variable_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 0035: 98@ >= 50@ // (float)

Sanny Builder SASCM.INI: 0035=2, %1d% >= %2d% ; (float)

Sanny Builder short use: 98@ >= 50@

Variable types must be declared before.

Description: Math opcode. Checks if specified Float value of the first Local variable is greater or equal to the Float value stored in second Local variable.

Opcode definition:

```
bool is_float_local_variable_greater_or_equal_to_local_variable (float
Local_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Local variable supposed as greater or equal
- 2) Passed: Float, Local variable variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0035_is_float_global_variable_greater_or_equal_to_global_variable
wait 0
0@ = 60.0
1@ = 50.0

    if
    0035: 0@ >= 1@ // (float)
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value of local
variable is greater or equal to the value of second global variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value of local
variable is lower than the value of second local variable" 2000
    end

jump 0035_is_float_global_variable_greater_or_equal_to_global_variable
```

0036 - is_float_global_variable_greater_or_equal_to_local_variable

Sanny Builder opcodes.txt: 0036: \$TEMPVAR_FLOAT_1 >= 181@(217@,8f) // (float)

Sanny Builder SASCM.INI: 0036=2, %1d% >= %2d% ; (float)

Sanny Builder short use: \$TEMPVAR_FLOAT_1 >= 50@

Variable types must be declared before.

Description: Math opcode. Checks if specified Float value of the first Global variable is greater or equal to the Float value stored in second Local variable.

Opcode definition:

```
bool is_float_global_variable_greater_or_equal_to_local_variable (float
Global_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Global variable supposed as greater or equal
- 2) Passed: Float, Local variable variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0036_is_float_global_variable_greater_or_equal_to_local_variable
wait 0
$TEMPVAR_FLOAT_1 = 60.0
1@ = 50.0

    if
    0036:  $TEMPVAR_FLOAT_1 >= 1@  // (float)
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is greater or equal to the value of second local variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value of global
variable is lower than the value of second local variable" 2000
    end

jump 0036_is_float_global_variable_greater_or_equal_to_local_variable
```

0037 - is_float_local_variable_greater_or_equal_to_global_variable

Sanny Builder opcodes.txt: 0037: 189@(217@,8f) >= \$TEMPVAR_FLOAT_2 // (float)

Sanny Builder SASCM.INI: 0037=2, %1d% >= %2d% ; (float)

Sanny Builder short use: 50@>= \$TEMPVAR_FLOAT_1

Variable types must be declared before.

Description: Math opcode. Checks if specified Float value of the first Local variable is greater or equal to the Float value stored in second Global variable.

Opcode definition:

```
bool is_float_global_variable_greater_or_equal_to_local_variable (float
Local_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Local variable supposed as greater or equal
- 2) Passed: Float, Global variable variable to compare first variable

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0037_is_float_local_variable_greater_or_equal_to_global_variable
wait 0
$TEMPVAR_FLOAT_1 = 50.0
l@ = 60.0

    if
    0037:  l@ >= $TEMPVAR_FLOAT_1  // (float)
    then
    0AD1: show_formatted_text_highpriority "specified FLOAT value of local
variable is greater or equal to the value of second global variable" time
2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified FLOAT value of local
variable is lower than the value of second global variable" 2000
    end

jump 0037_is_float_local_variable_greater_or_equal_to_global_variable
```

0038 - is_int_global_variable_equal_to_value

Sanny Builder opcodes.txt: 0038: \$672 == 1

Sanny Builder SASCM.INI: 0038=2, %1d% == %2d%

Sanny Builder short use: \$672 == 1

Description: Math opcode. Checks if the Integer value of Global variable is equal to the specified value.

Opcode definition:

```
bool is_int_global_variable_equal_to_value (int Global_variable, int
```

Parameters:

- 1) Passed: Integer, Global variable
- 2) Passed: Integer, Value to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0038_is_int_global_variable_equal_to_value
wait 0
$89 = 2

    if
    0038:  $89 == 2
    then
    0AD1: show_formatted_text_highpriority "specified INT value of global
variable is equal to the value" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified INT value of global
variable is not equal to the value" 2000
    end

jump 0039_is_int_global_variable_equal_to_value
```

0039 - is_int_local_variable_equal_to_value

Sanny Builder opcodes.txt: 0039: 1@ == 0

Sanny Builder SASCM.INI: 0037=2, %1d% >= %2d% ; (float)

Sanny Builder short use: 1@ == 0

Description: Math opcode. Checks if the Integer value of Local variable is equal to the specified value.

Opcode definition:

```
bool is_int_local_variable_equal_to_value (int Local_variable, int
```

Parameters:

- 1) Passed: Integer, Local variable
- 2) Passed: Integer, Value to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0039_is_int_local_variable_equal_to_value
wait 0
l@ = 0

    if
    0039:  l@ == 0
    then
    0AD1: show_formatted_text_highpriority "specified INT value of local
variable is equal to the value" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "specified INT value of local
variable is not equal to the value" 2000
    end

jump 0039_is_int_local_variable_equal_to_value
```

003A - is_int_global_variable_equal_to_global_variable

Sanny Builder opcodes.txt: 003A: \$GIRL_DATED_NOW == \$GIRLFRIEND //
(int)

Sanny Builder SASCM.INI: 003a=2, %1d% == %2d% ; (int)

Sanny Builder short use: \$GIRL_DATED_NOW == \$GIRLFRIEND

Variable types must be declared before.

Description: Math opcode. Checks if the Integer value of Local variable is equal to the value stored in second Local variable.

Opcode definition:

```
bool is_int_global_variable_equal_to_global_variable (int
Global_variable, int Global_variable)
```

Parameters:

- 1) Passed: Integer, Global variable
- 2) Passed: Integer, Global variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:003A_is_int_global_variable_equal_to_global_variable
wait 0
$89 = 50
$90 = 50

if
003A: $89 == $90 // (int)
then
0AD1: show_formatted_text_highpriority "INT value of global variable is
equal to the value stored in second global variable" time 2000 // It will
appear
else
0AD1: show_formatted_text_highpriority "INT value of global variable is
not equal to the value stored in second global variable" 2000
end

// 003A_is_int_global_variable_equal_to_global_variable
```

003B - is_int_local_variable_equal_to_local_variable

Sanny Builder opcodes.txt: 003B: 18@ == 21@ // (int)

Sanny Builder SASCM.INI: 003b=2, %1d% == %2d% ; (int)

Sanny Builder short use: 18@ == 21@

Variable types must be declared before.

Description: Math opcode. Checks if the Integer value of Local variable is equal to the value stored in second Local variable.

Opcode definition:

```
bool is_int_local_variable_equal_to_local_variable (int
Local_variable, int Local_variable)
```

Parameters:

- 1) Passed: Integer, Local variable
- 2) Passed: Integer, Local variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:003B_is_int_local_variable_equal_to_local_variable
wait 0
18@ = 50
21@ = 50

if
003B: 18@ == 21@ // (int)
then
0AD1: show_formatted_text_highpriority "INT value of local variable is
equal to the value stored in second local variable" time 2000 // It will
appear
else
0AD1: show_formatted_text_highpriority "INT value of local variable is
not equal to the value stored in second local variable" 2000
end

// 003B_is_int_local_variable_equal_to_local_variable
```

003C - is_int_global_variable_equal_to_local_variable

Sanny Builder opcodes.txt: 003C: \$CAR_MODELS_TO_EXPORT(4@,10i) == 6@
// (int)

Sanny Builder SASCM.INI: 003c=2, %1d% == %2d% ; (int)

Sanny Builder short use: \$CAR_MODELS_TO_EXPORT(4@,10i) == 6@

Variable types must be declared before.

Description: Math opcode. Checks if the Integer value of Local variable is equal to the value stored in second Local variable.

Opcode definition:

```
bool is_int_global_variable_equal_to_local_variable (int
Global_variable, int Local_variable)
```

Parameters:

- 1) Passed: Integer, Global variable
- 2) Passed: Integer, Local variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:003C_is_int_global_variable_equal_to_local_variable
wait 0
6@ = 50
$89 = 50

if
003C: $89 == 6@ // (int)
then
0AD1: show_formatted_text_highpriority "INT value of global variable is
equal to the value stored in second local variable" time 2000 // It will
appear
else
0AD1: show_formatted_text_highpriority "INT value of global variable is
not equal to the value stored in second local variable" 2000
end

// 003C_is_int_global_variable_equal_to_local_variable
```

0042 - is_float_global_variable_equal_to_value

Sanny Builder opcodes.txt: 0042: \$279 == 0.0

Sanny Builder SASCM.INI: 0042=2, %1d% == %2d%

Sanny Builder short use: \$279 == 0.0

Description: Math opcode. Checks if the Float value of Global variable is equal to the specified value.

Opcode definition:

```
bool is_float_global_variable_equal_to_value (float Global_variable,
float Value)
```

Parameters:

- 1) Passed: Float, Global variable
- 2) Passed: Float, Value to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0042_is_float_global_variable_equal_to_value
wait 0
$TEMPVAR_FLOAT_1 = 52.0

    if
    0042: $TEMPVAR_FLOAT_1 == 52.0
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is equal to the specified value" time 2000 // It will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is not equal to the specified value" 2000
    end

jump @0042_is_float_global_variable_equal_to_value
```

0043 - is_float_local_variable_equal_to_value

Sanny Builder opcodes.txt: 0043: 301@ == 0.2

Sanny Builder SASCM.INI: 0043=2, %1d% == %2d%

Sanny Builder short use: 301@ == 0.2

Description: Math opcode. Checks if the Float value of Local variable is equal to the specified value.

Opcode definition:

```
bool is_float_local_variable_equal_to_value (float Local_variable,
float Value)
```

Parameters:

- 1) Passed: Float, Local variable
- 2) Passed: Float, Value to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }  
// Example requires CLE04  
0000: NOP  
  
:0043_is_float_local_variable_equal_to_value  
wait 0  
0@ = 0.2  
  
    if  
    0043:    0@ == 0.2  
    then  
    0AD1: show_formatted_text_highpriority "FLOAT value of local variable is  
equal to the specified value" time 2000 // It will appear  
    else  
    0AD1: show_formatted_text_highpriority "FLOAT value of local variable is  
not equal to the specified value" 2000  
    end  
  
iump @0043 is float local variable equal to value
```

0044 - is_float_global_variable_equal_to_global_variable

Sanny Builder opcodes.txt: 0044: \$3499 == \$3507(\$8549,151f) //
(float)

Sanny Builder SASCM.INI: 0044=2, %1d% == %2d% ; (float)

Sanny Builder short use: \$3499 == \$3507(\$8549,151f)

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of the first Global variable is equal to the value stored in second Global variable.

Opcode definition:

```
bool is_float_global_variable_equal_to_global_variable (float  
Global_variable, float Global_variable)
```

Parameters:

- 1) Passed: Float, Global variable
- 2) Passed: Float, Global variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0044_is_float_global_variable_equal_to_global_variable
wait 0
$TEMPVAR_FLOAT_1 = 4.5
$TEMPVAR_FLOAT_2 = 4.5

    if
    0044: $TEMPVAR_FLOAT_1 == $TEMPVAR_FLOAT_2 // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is equal to the value stored in second global variable" time 2000 // It
will appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is not equal to the value stored in second global variable" 2000
    end

jump 0044_is_float_global_variable_equal_to_global_variable
```

0045 - is_float_local_variable_equal_to_local_variable

Sanny Builder opcodes.txt: 0045: 85@ == 69@ // (float)

Sanny Builder SASCM.INI: 0045=2, %1d% == %2d% ; (float)

Sanny Builder short use: 85@ == 69@

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of the first Local variable is equal to the value stored in second Local variable.

Variable types must be declared before.

Opcode definition:

```
bool is_float_local_variable_equal_to_local_variable (float
Local_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Local variable
- 2) Passed: Float, Local variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0045_is_float_local_variable_equal_to_local_variable
wait 0
0@ = 0.1
1@ = 0.1

    if
    0045: 0@ == 1@ // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of local variable is
equal to the value stored in second local variable" time 2000 // It will
appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of local variable is
not equal to the value stored in second local variable" 2000
    end

jump 0045_is_float_local_variable_equal_to_local_variable
```

0046 - is_float_global_variable_equal_to_local_variable

Sanny Builder opcodes.txt: 0046: \$var == 0@ // (float)

Sanny Builder SASCM.INI: 0046=2, %1d% == %2d% ; (float)

Sanny Builder short use: \$TEMPVAR_FLOAT_1 == 0@

Variable types must be declared before.

Description: Math opcode. Checks if the Float value of the first Global variable is equal to the value stored in second Local variable.

Variable types must be declared before.

Opcode definition:

```
bool is_float_global_variable_equal_to_local_variable (float
Global_variable, float Local_variable)
```

Parameters:

- 1) Passed: Float, Global variable
- 2) Passed: Float, Local variable to compare first variable with

Returns true or false? Yes.

Example in Sanny Builder:

```
{ $CLE0 }
// Example requires CLE04
0000: NOP

:0046_is_float_global_variable_equal_to_local_variable
wait 0
$TEMPVAR_FLOAT_1 = 0.1
0@ = 0.1

    if
    0046: $TEMPVAR_FLOAT_1 == 0@ // (float)
    then
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is equal to the value stored in second local variable" time 2000 // It will
appear
    else
    0AD1: show_formatted_text_highpriority "FLOAT value of global variable
is not equal to the value stored in second local variable" 2000
    end

jump 0046_is_float_global_variable_equal_to_local_variable
```

004D - jump_if_false

Sanny Builder opcodes.txt: 004D: jump_if_false @MAIN_4068

Sanny Builder SASCM.INI: 004D=1,jump_if_false %1p%

Sanny Builder keyword: else_jump (else_jump @label)
 jf (jf @label)

Description: Jumps the code to the specified offset called label if "false" is the current state of the thread. Label is negative number, offset relative to beginning of SCM file.

Opcode definition:

```
void jump_if_false (int Offset)
```

Parameters:

1) Passed: integer, label - negative offset in script file. (OFFSET * -

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }  
// Example requires CLE04  
0@ = 3  
  
:004D_jump_if_false  
wait 0  
if  
0@ == 3  
004D: jump_if_false @not_3  
0AD1: show_formatted_text_highpriority "variable contains 3 value" time 2000  
jump @004D_jump_if_false  
  
:not_3  
0AD1: show_formatted_text_highpriority "variable does not contain 3 value"  
time 2000  
jump @004D_jump_if_false
```

004E - TERMINATE_THIS_SCRIPT

Sanny Builder opcodes.txt: 004E: `end_thread`

Sanny Builder SASCM.INI: 004e=0,end_thread

Sanny Builder keyword: `end_thread`

Native name: TERMINATE_THIS_SCRIPT

Description: Terminates current thread, thus current thread is no longer executed .

Opcode definition:

```
void terminate_this_script ()
```

Parameters:

No parameters

Returns true or false? No.

Example in Sanny Builder:

```
// It is not stripped SCM to reuse
// Many things are missing here
// It is just example

// Instruction:
// Compile this main.txt as main.scm
// Run a new game
// In game GXT text will be shown
// Press entering key, enter or F to terminate this thread
// GXT text will disappear

DEFINE MISSIONS 0
//DEFINE MISSION {ID} 0 AT {LABEL} @
DEFINE EXTERNAL_SCRIPTS 0 // Use -1 in order not to compile AAA script
//DEFINE SCRIPT {NAME} AT {LABEL} @
DEFINE UNKNOWN_EMPTY_SEGMENT 0
DEFINE UNKNOWN_THREADS_MEMORY 0

0053: $PLAYER_CHAR = create_player #NULL at 2488.56 -1666.84 13.38
01F5: $PLAYER_ACTOR = create_player_actor $PLAYER_CHAR
07AF: $PLAYER_GROUP = player $PLAYER_CHAR group
070D: rebuild_player $PLAYER_CHAR
016A: fade 1 time 0
0180: set_on_mission_flag_to $ONMISSION

// put your create_thread commands here
CREATE_THREAD @004E_end_thread // Thread will be executed since now
(after first wait from this thread)
// More threads will be active

:MAIN_LOOP
wait 0
jump @MAIN_LOOP

:004E_end_thread
wait 0
if
00E1: player 0 pressed_key 15 // Press Enter / Exit and this thread
will be terminated
else_jump @Show_GXT_text

004E: end_thread // This thread is terminated and is not executed
anymore. GXT text will disappear

:Show_GXT_text
00BC: show_text highpriority GXT 'TF23' time 1000 flag 1
```

004F - START_NEW_SCRIPT_WITH_ARGS

Sanny Builder opcodes.txt: 004F: create_thread @MS_BIKE_MISSIONS

Sanny Builder SASCM.INI: 004f=-1,create_thread

Sanny Builder keyword: `create_thread` (`create_thread @thread_label`)

Description: Starts new thread pointing to a specified label . Should be used in main.scm, not in CLEO scripts. Created thread will be need own `wait` opcodes and jumps, because it works separately.

Native name: START_NEW_SCRIPT_WITH_ARGS

Opcode definition:

```
void start_new_script (int Offset)
```

Parameters:

```
1) Passed: integer, label - negative offset in script file. (OFFSET * -
```

Returns true or false? No.

Example in Sanny Builder:

```
// It is not stripped SCM to reuse
// Many things are missing here
// It is just example

// Instruction:
// Compile this main.txt as main.scm
// Start a new game
// In game GXT text will be shown constantly

DEFINE MISSIONS 0
//DEFINE MISSION {ID} 0 AT {LABEL} @
DEFINE EXTERNAL_SCRIPTS 0 // Use -1 in order not to compile AAA script
//DEFINE SCRIPT {NAME} AT {LABEL} @
DEFINE UNKNOWN_EMPTY_SEGMENT 0
DEFINE UNKNOWN_THREADS_MEMORY 0

0053: $PLAYER_CHAR = create_player #NULL at 2488.56 -1666.84 13.38
01F5: $PLAYER_ACTOR = create_player_actor $PLAYER_CHAR
07AF: $PLAYER_GROUP = player $PLAYER_CHAR group
070D: rebuild_player $PLAYER_CHAR
016A: fade 1 time 0
0180: set_on_mission_flag_to $ONMISSION

// put your create_thread commands here
004F: create_thread @Show_GXT_text_show // Thread will be executed
since now (after first wait from this thread)
// More threads will be active

:MAIN_LOOP
wait 0
jump @MAIN_LOOP

:Show_GXT_text_show
wait 0
00BC: show_text_highpriority GXT 'IE23' time 1000 flag 1
:GXT_finished
```

0050 - gosub

Sanny Builder opcodes.txt: 0050: gosub @SUB_FADE_500MS

Sanny Builder SASCM.INI: 0050=1,gosub %1p%

Sanny Builder keyword: gosub (gosub @label)

Description: Executes the code in the specified label until the label returns. Then the code carries on.

Opcode definition:

```
void gosub (int Offset)
```

Parameters:

1) Passed: integer, label - negative offset in script file. (OFFSET * -

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP

{
  Instruction:
  - Compile CLE0 script
  - Run the game
  - Press Enter/F to execute gosub
  - Player will be locked or unlocked
}

:0050_gosub_lock
wait 0
if
00E1: player 0 pressed_key 15 // Press Enter/F
else_jump @0050_gosub_lock

      0050: gosub @PLAYER_LOCK

wait 400

:0050_gosub_unlock
wait 0
if
00E1: player 0 pressed_key 15 // Press Enter/F
else_jump @0050_gosub_unlock

      0050: gosub @PLAYER_RESTORE_CONTROL

wait 400
}
```

0051 - return

Sanny Builder opcodes.txt: 0051: return

Sanny Builder SASCM.INI: 0051=0,return

Sanny Builder keyword: return

Description: Returns the code back to where the gosub opcode was used.

Opcode definition:

void gosub ()

Parameters:

No parameters

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLEO }
0000: NOP

{
  Instruction:
    - Compile CLEO script
    - Run the game
    - Press Enter/F to execute gosub
    - Player will be locked or unlocked
}

:0050_gosub_lock
wait 0
if
00E1: player 0 pressed_key 15      // Press Enter/F
else_jump @0050_gosub_lock

      gosub @PLAYER_LOCK

wait 400

:0050_gosub_unlock
wait 0
if
00E1: player 0 pressed_key 15      // Press Enter/F
else_jump @0050_gosub_unlock

      gosub @PLAYER_RESTORE_CONTROL

wait 400
jump @0050_gosub_lock

:PLAYER_LOCK
Actor.LockInCurrentPosition($PLAYER_ACTOR) = True
Actor.SetImmunities($PLAYER_ACTOR, 1, 1, 1, 1, 1)
Player.CanMove($PLAYER_CHAR) = False
0051: return      // described opcode here

:PLAYER_RESTORE_CONTROL
Player.CanMove($PLAYER_CHAR) = True
Actor.SetImmunities($PLAYER_ACTOR, 0, 0, 0, 0, 0)
Actor.LockInCurrentPosition($PLAYER_ACTOR) = False
```

0052 - NOP_floats

Sanny Builder opcodes.txt: 0052: NOP 98@ 100@ \$TEMPVAR_FLOAT_3 99@ 100@ \$TEMPVAR_FLOAT_3

Sanny Builder SASCM.INI: 0052=6,NOP %1d% %2d% %3d% %4d% %5d% %6d%

Description: Does nothing. This opcode probably showed values of arguments on screen at development stage.

Opcode definition:

```
void NOP_floats (float Argument_1, float Argument_2, float Argument_3, float Argument_4, float Argument_5, float Argument_6)
```

Parameters:

- 1) Passed: float, argument 1
- 2) Passed: float, argument 2
- 3) Passed: float, argument 3
- 4) Passed: float, argument 4
- 5) Passed: float, argument 5
- 6) Passed: float, argument 6

Returns true or false? No.

0053 - CREATE_PLAYER

Sanny Builder opcodes.txt: 0053: \$PLAYER_CHAR = create_player #NULL at 2488.562 -1666.865 12.8757

Sanny Builder SASCM.INI: 0053=5,%5d% = create_player %1o% at %2d% %3d% %4d%

Sanny Builder class method: Player.Create

Native name: CREATE_PLAYER

Description: Creates player at the specified coordinates. The model works differently to other creation opcodes:

#NULL - Player 1

#CSPLAY - Player 2

Player 1 wears the default CJ skin with the default clothes.

Player 2 will have exactly the same model and clothes as the current set for Player 1. Creating Player 2 will create another HUD showing health, armour etc. of the second player.

To change the clothes of any player, use opcode 087B. The players model can be changed using 097C.

Creating a second player with the model #NULL creates a duplicate of the player which can never be destroyed (using 06DF results in a crash).

Opcode definition:

```
void create_player (int Model_ID, float X_coord, float Y_coord, float Z_coord, &int Player_handle)
```

Parameters:

- 1) Passed: integer, Model/Ped ID
- 2) Passed: float, X coordinate
- 3) Passed: float, Y coordinate
- 4) Passed: float, Z coordinate
- 5) Stored: integer, Actor handle

Returns true or false? No.

Example in Sanny Builder:

```
// It is not stripped SCM to reuse
// Many things are missing here
// It is just example

// Instruction:
// Compile this main.txt as main.scm
// Start a new game

DEFINE MISSIONS 0
//DEFINE MISSION {ID} 0 AT {LABEL} @
DEFINE EXTERNAL_SCRIPTS 0 // Use -l in order not to compile AAA script
//DEFINE SCRIPT {NAME} AT {LABEL} @
DEFINE UNKNOWN_EMPTY_SEGMENT 0
DEFINE UNKNOWN_THREADS_MEMORY 0

0053: $PLAYER_CHAR = create_player #NULL at 2488.56 -1666.84 13.38 //
described opcode here
// Player must be created before a first instance of wait opcode
070D: rebuild_player $PLAYER_CHAR

01F5: $PLAYER_ACTOR = create_player_actor $PLAYER_CHAR
07AF: $PLAYER_GROUP = player $PLAYER_CHAR group
016A: fade 1 time 0

end thread
```

0058 - add_int_global_variable_to_global_variable

Sanny Builder opcodes.txt: 0058: \$1924 += \$1929 // (int)

Sanny Builder SASCM.INI: 0058=2,%1d% += %2d% ; (int)

Sanny Builder short use: \$1924 += \$1929

Variable types must be declared before.

Description: Math opcode. Adds the integer value of the second Global Variable to the integer value of the first Global Variable.

Opcode definition:

```
void add_int_global_variable_to_global_variable (int &Global_variable,
int Global_variable)
```

Parameters:

- 1) Stored: integer, global variable
- 2) Passed: integer, global variable to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP
$89 = 40
$90 = 50
0058: $89 += $90 // (int)

:0058_add_global_variable_to_global_variable
01E4: show_text_1number_lowpriority GXT 'NUMBER' number $89 time 2000 flag 1
wait 1000
jump @0058 add global variable to global variable
```

0059 - add_float_global_variable_to_global_variable

Sanny Builder opcodes.txt: 0059: \$1316 += \$1317 // (float)

Sanny Builder SASCM.INI: 0059=2,%1d% += %2d% ; (float)

Sanny Builder short use: \$1316 += \$1317

Variable types must be declared before.

Description: Math opcode. Adds the float value of the second Global Variable to the float value of the first Global Variable.

Opcode definition:

```
void add_float_global_variable_to_global_variable (float
&Global_variable, float Global_variable)
```

Parameters:

- 1) Stored: float, global variable
- 2) Passed: float, global variable to add

Returns true or false? No.**Example in Sanny Builder:**

```

{$CLE0}
0000: NOP
$TEMPVAR_FLOAT_1 = 1.5
$TEMPVAR_FLOAT_2 = 0.5

0059: $TEMPVAR_FLOAT_1 += $TEMPVAR_FLOAT_2 // (float)
// Result: 2.0

:0059_add_float_global_variable_to_global_variable
0AD1: show_formatted_text_highpriority "float value of global variable: %.3f"
time 2000 $TEMPVAR_FLOAT_1
wait 0
jump @0059_add_float_global_variable_to_global_variable

```

005A - add_int_local_variable_to_local_variable**Sanny Builder opcodes.txt:** 005A: 3@ += 1@ // (int)**Sanny Builder SASCM.INI:** 005a=2,%1d% += %2d% ; (int)**Sanny Builder short use:** 3@ += 1@

Variable types must be declared before.

Description: Math opcode. Adds the integer value of the second Local Variable to the integer value of the first Local Variable.**Opcode definition:**

```
void add_int_local_variable_to_local_variable (int &Local_variable, int Local_variable)
```

Parameters:

- 1) Stored: integer, local variable
- 2) Passed: integer, local variable to add

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
0000: NOP
0@ = 4
1@ = 2
005A: 0@ += 1@ // (int)
// Result: 6

:005A_add_int_local_variable_to_local_variable
01E4: show_text_1number_lowpriority GXT 'NUMBER' number 0@ time 2000 flag 1
wait 1000
jump @005A_add_int_local_variable_to_local_variable
```

CLE03 opcodes

0A8C - write_memory

Sanny Builder opcodes.txt: 0AB0: key_pressed 0x73

Sanny Builder SASCM.INI: 0AB0=1, key_pressed %1d%

Description: Writes a value to the game memory.

Opcode definition:

```
boolean virtual_key_pressed (int Memory_address, int Size, int Value,
boolean Virtual_protect)
```

Parameters:

- 1) Passed: Integer, Memory adress
- 2) Passed: Integer, Number of bytes from value to write: 1, 2 or 4-bytes
- 3) Passed: Integer, Value to write

Returns true or false? No.

Example in Sanny Builder:

```
{ $CLE0 }
// Draws scanlines on screen
// Works for GTA San Andreas v1.0 [US] HOODLUM No-CD Fixed EXE
0A8C: write_memory 0x0C7C70C size 1 value 1 virtual_protect 0
0A8C: write_memory 0x0C7C70D size 1 value 1 virtual_protect 0
```

0AB0 - virtual_key_pressed

Sanny Builder opcodes.txt: 0AB0: key_pressed 0x73

Sanny Builder SASCM.INI: 0AB0=1, key_pressed %1d%

Sanny Builder class method: **Key.VirtualKeyCode**

Description: This opcode tests if the key is pressed on keyboard. If the key with specified code is pressed, it returns True, otherwise False.

See table of

-
-
-
-
-

[illegible]

Terminology

Obj – object

Opcode native name – original names of opcodes used by Rockstar Games to write their scripts, retrieved from GTA IV which uses “native functions”.

SB – Sanny Builder, program created by Seemann designed for the GTA 3D game series (GTA3, VC, SA; partially LCS and VCS).

It includes a decompiler, permitting the end-user to quickly decompile the MAIN.SCM file which contains game scripts. The compiler feature offers a convenient editor with a large number of useful functions such as; syntax highlighting, error checking, advanced search tools, player coordinates reading, fast movement through code and much more.

SCM – extension of most important file in scripting – main.scm, this file contains missions and scripts.

VTOL - acronym for vertical take-off and landing aircraft. This classification includes fixed-wing aircraft that can hover, take off and land vertically as well as helicopters and other aircraft with powered rotors, such as tiltrotors. The terminology for spacecraft and rockets is VTVL (vertical takeoff with vertical landing). Some VTOL aircraft can operate in other modes as well, such as CTOL (conventional take-off and landing), STOL (short take-off and landing), and/or STOVL (short take-off and vertical landing). Others, such as some helicopters, can only operate by VTOL, due to the aircraft lacking landing gear that can handle horizontal motion. VTOL is a subset of V/STOL (vertical and/or short take-off and landing). Hydra in GTA SA is VTOL aircraft.

References

1. EnbSeries documentation: http://enbdev.com/doc_en.htm
2. Virtual key codes: [http://msdn.microsoft.com/en-us/library/dd375731\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/dd375731(v=vs.85).aspx)
3. Virtual key codes: http://www.kbdedit.com/manual/low_level_vk_list.html
4. ASCII codes: <http://www.ascii-code.com/>
5. Definition of paths.ipl for GTA Vice City: <http://projectcerbera.com/gta/vc/tutorials/paths>

Book author: **fastman92**